On-line Surface Roughness Estimation in Cylindrical Turning Using Neural Networks

Brent Andrew Backhouse

B.E (Mech. Hons)

Submitted in fulfillment of the requirements for the degree of

Masters of Engineering Science (M.Eng.Sc)

Civil and Mechanical Engineering

UNIVERSITY OF TASMANIA

November 2000

STATEMENT OF ORIGINALITY AND AUTHORITY OF ACCESS

This thesis contains no material that has been accepted for a degree or diploma by The University of Tasmania or any other institution, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgement is made in the text of the thesis.

This thesis may be made available for loan and limited copying in accordance with the *Copyright Act 1968.*

Brent Andrew Backhouse

ABSTRACT

In recent years a direct method of surface finish quality detection by electrical resistance, optical, image processing and dial indicator methods have been proved to be quantitatively unreliable. The cutting tool wear undergoes a gradual increase and the failure at the end of useful life is decided on the extent of the wear growth. This irregular tool wear trend proportionally causes the quality of the surface finish to be unpredictable. Therefore there is a great need for a reliable quantitative method to establish the quality of a work materials surface finish, online.

From an industry point of view, it is often necessary to identify the production quality of work materials to avoid expensive losses on machines. A worn or wearing cutting tool will cause a gradual decline in the surface quality of a component, causing unfavorable circumstances. For example, machining of alloy wheels in Ford, Chrysler and GMH have a specific surface roughness to achieve for effective coating of the wheels. The quality of the wheel coatings and their 'scratch resistance' depend on the surface profile produced during the machining operation. While many industries adopt a 'direct modeling' approach, where the machining process variables are given as inputs to the process to estimate the surface roughness, there is no evidence available of the 'inverse modeling' where the operating conditions are estimated based on the target values of the surface roughness. Nevertheless, using the direct modeling the surface roughness can only be measured 'off line' as a quality control exercise to meet the specifications of the wheel. There is a general disagreement with the manufacturer specifications of the surface roughness, on line, with the constant changes in the cutting tool conditions and wear. Hence there is a need for 'on line' determination of the surface roughness while carrying out the machining operation. This will give an indication on the extent of tool wear growth. Further, this study will also give quantitative values of surface roughness 'on line' and its application to the manufacturing process.

In this study a neural network model is proposed for surface roughness detection 'on-line' in cylindrical turning operations. Neural network architectures will be used as 'direct' modeling techniques to estimate the surface roughness. As 'inverse'

modeling tools the operating conditions will also be predicted using the surface roughness and cutting tool wear as inputs.

Extensive turning experiments on a lathe will be carried out covering a comprehensive range of cutting conditions to generate the knowledge base for the training stage of the neural network algorithms. The process parameters measured during the experimentation for identification of surface finish quality includes the forces, cutting tool vibrations and surface finish during cylindrical turning operation. These problem addresses one of the most pressing needs of modern automobile manufacturing industry where an 'on line' estimation of surface roughness is seen as an important parameter.

This project while improving the understanding of the machining parameters and their influence on surface finish will also identify necessary neural network tools for application. This work is seen as a step towards establishing intelligent tools for machining performance estimation, while addressing the mathematical and scientific basis of machining science.

ACKNOWLEDGEMENTS

Firstly I would like to thank my supervisor Dr Vishy Karri for not only his technical support and guidance, but his friendship throughout the degree. Without his professional approach none of this would have been possible.

My family, who have made it possible for me to be here, not just financially but through their love and caring support.

All the staff of the Engineering faculty who have guided me in my six years with the university.

A special mention to the workshop staff for their patience and guidance throughout my experimentation.

Finally, to all my friends who have picked me up when I was down and shown me back onto the path.

ADDENDUM

Page 17 - In the heading of section 2.2 and 2.3 monitoring is misspelt.

Page 28 - In the heading of section 2.4 monitoring is misspelt.

Chapter 6 - The following list of R^2 values should be included with the corresponding figure

Figure Number	Output	R^2 Value
Figure 6.1	Actual Feed	0.988
	Actual Tang	0.896
Figure 6.2	Frequency	0.909
Figure 6.3	Ra	0.940
	Ry	0.468
Figure 6.5	Ffeed	0.014
	Ftang	0.004
Figure 6.6	Freq	0.232
Figure 6.7	Ra	0.779
	Ry	0.697
Figure 6.9	Ffeed	0.941
	Ftang	0.939
Figure 6.10	Freq	0.894
Figure 6.11	Ra	0.755
	Ry	0.709
Figure 6.12	PF tool Feed	0.988
	PF tool Tang	0.897
	PM tool Feed	0.945
	PM tool Tang	0.935
Figure 6.13	PF tool Freq	0.909
	PM tool Freq	0.265
Figure 6.14	PF tool Ra	0.940
	PM tool Ra	0.806

Figure Number	Output	R^2 Value
Figure 6.15	PF tool Ry	0.468
	PM tool Ry	0.778
Figure 6.16	PF tool Feed	0.014
	PM tool Feed	0.223
Figure 6.17	PF tool Tang	0.004
	PM tool Tang	0.036
Figure 6.18	PF tool Freq	0.232
	PM tool Freq	0.324
Figure 6.19	PF tool Ra	0.779
	PM tool Ra	0.921
Figure 6.20	PF tool Ry	0.697
	PM tool Ry	0.923
Figure 6.21	PF tool Feed	0.941
	PF tool Tang	0.939
	PM tool Feed	0.974
	PM tool Tang	0.939
Figure 6.22	PF tool Freq	0.894
	PM tool Freq	0.666
Figure 6.23	PF tool Ra	0.755
	PM tool Ra	0.906
Figure 6.24	PF tool Ry	0.709
	PM tool Ry	0.263

Page 228 - In the first line "PR" should read "PF".

Appendix D - The following list of R^2 values should be included with the corresponding figure

Figure Number	Output	R^2 Value
Figure D.1	Actual Feed	0.945
	Actual Tang	0.935
Figure D.2	Frequency	0.265
Figure D.3	Ra	0.806
	Ry	0.778
Figure D.4	Ffeed	0.223
	Ftang	0.036
Figure D.5	Frequency	0.324

Figure Number	Output	R^2 Value
Figure D.6	Ra	0.921
	Ry	0.923
Figure D.7	Ffeed	0.974
	Ftang	0.939
Figure D.8	Freq	0.666
Figure D.9	Ra	0.906
	Ry	0.263

CONTENTS

CHAPTER 1

INTRODUCTION

Machining operations have in recent times gained much importance due to the inherent advantages of technology. Since ancient times machining practices have been used to manufacture items required for everyday life. Furthermore now, with the increase in production and demand for components of such a wide spectrum, for all facets of society. The modern world is reliant on material removal processes, for mechanical purposes, such as the automotive industry, where most of a cars precision parts, are created through machining processes. Also for structural purposes, the many thousand tons of metal used to create a new building development, have been machined in one way or another to suit their particular function.

The machine tool conditions and its effect on surface quality, has been a major concern as it is unavoidable, particularly so when absolute failure is unpredictable. It is a costly affair when produce are not made to the quality specifications anymore when such wear and tear on machine ageing and tool wear and breakage happens. The cost associated with downgrade of quality and downtime due to wear and breakage may at first seem negligible, but express this hundreds or even thousands of times over, the accumulating cost factor is certainly worrying. This is even more so if the cost of production is high and yield is low.

Therefore with the increasing demand for output from a production process, and the rapid improvement in technology, there is a vast need for performance estimation. Performance estimation relates the variables of the machining process to the output quality achieved. In the past, less respect has been given to the efficiency of the process and the product of quality achieved. With changing times, and the demand for a higher quality, more durable product, it is imperative both the product quality and the process to achieve the product are highly monitored. The quality of components demanded by customers is much greater, and therefore the demand for

lower irregularities in the surface finish quality is very important. No direct accurate methods have been able to resolve this problem to date. This is what this project hopes to achieve.

The literature review following gives an in depth discussion of the previous monitoring techniques used previously. These start from the classical orthogonal cutting and empirical approaches, used to give a basic understanding of the machining variables, and then discuss previous qualitative and quantitative investigations into this field. With the increase in technology and the online implementation of computer technology, it is now feasible to almost fully automate the performance estimation. What this means is that with smart computer aided devices such as fuzzy logic, or Neural Networks, the material removal process can be monitored (all its variables) online. Thus given a desired quality standard target of the finished work piece, the feed, speed, and depth of cut can be regulated by the "smart system" to obtain the required quality.

Performance prediction of a manufacturing process occurs in two different ways: direct and inverse modeling. [1] Direct modeling is the most widely used to date. The method uses the process variables as the input to predict the performance of the system. What this involves is, given that the forces, surface finish, temperature, etc, of the system are known, information about the tool material, work material, and geometry can be modeled or obtained. Inverse modeling gives industry an alternative, whereby, given a required level of surface finish, the input parameters are required. Therefore given that feed, speed, depth of cut, geometry, and tool material are known as inputs, the forces, surface finish, tool wear, and tool life can be modeled as output variables.

This thesis aims to accurately determine the state of the surface finish of the work piece, online and from this, a quantitative estimation of the life of the cutting tool. By firstly testing the wear parameters of the tool, through the variance of feed, speed, and depth of cut, and there corresponding relation to the forces present (two components), vibrations, surface finish of the work piece. Given that a relation between these parameters can be found, a smart system, namely the neural network, can be trained with the testing results previously achieved. The testing results will

come from a broad range of tests and a range of cutting tools, so the neural network will be trained to universally predict the surface finish profile over a wide range of cutting conditions. This will give the advantage that the wear prediction will be far more accurate than previous performance estimation techniques, and recommended manual guidelines.

Given that the online system can detect the surface finish of the work material, to a precise time period, the neural network software can be implemented to comply with the latest technology. Occurrence of this is in the automatic tool changing process (robotics) where the change time between the new and old tool is almost negligible compared to current methods.

This technology gives industry the solution that has been a concern and a great cost for so many years. It puts to an end the large wastage of work material, and work time, and thus at the end of the day, it means money saved. Not only does it save time and wastage, but, it ensures that a better product quality is continually achieved, through a more streamlined manufacturing approach.

The next chapter will review the literature and the specific areas of previous investigation in this field of research.

CHAPTER 2

LITERATURE REVIEW

This literature review describes the performance estimation techniques employed by previous researchers. Traditionally modeling for metal turning has been carried out using a direct approach, this being the mechanics of cutting approach. Although this work is aimed at estimating surface finish, a general outline of the various methods will be discussed below.

2.1 MECHANICS OF CUTTING

This part of the literature survey is dedicated to the mechanics of cutting. This describes the direct modeling process of turning mechanics.

2.1.1 Turning

Turning in its basic form, it can be defined as the machining of an external surface [2] [3]:

- with the workpiece rotating,
- with a single-point cutting tool, and
- with the cutting tool feeding parallel to the axis of the workpiece and at a distance that will remove the outer surface of the work.

Thus, in general for turning, the primary cutting motion is rotational (as opposed to linear transverse for an operation like shaping and planning). This motion is displayed in figure 2.1.

Variating slightly from the above definition, taper turning is where the cutter path is at an angle to the work axis. In contour turning, the distance of the cutter from the work axis is varied to produce the desired shape.

Turning operations may incorporate a single-point tool, however can also involve multiple-tool setups, which are often employed in turning. In such setups, each tool operates independently as a single-point cutter.

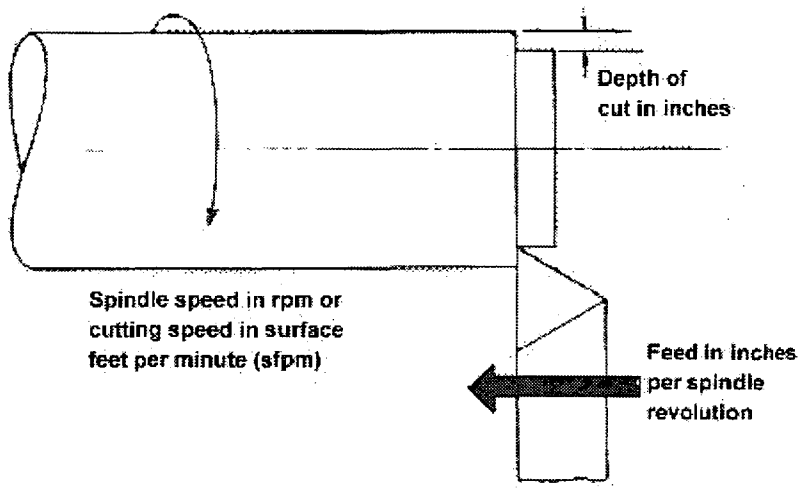

Figure 2.1 Typical turning operation [2]

The three primary factors in any basic turning operation are speed, feed, and depth of cut. Given these primary factors the cutting velocity of the tool with respect to the workpiece, V_w , can be found through vector addition of the tangential cutting speed:

$$V_w = \sqrt{\left(V^2 + V_f{}^2\right)} = \sqrt{\left(\pi N D^2 + f N^2\right)}$$

where:

V = tan *gental cutting speed*

D = *work piece diameter at the po*int *on the cutting edge* [2.1]

f = *feed*

N = *work piece revolutions per unit time*

V_f = *velocity of feed*

In most situations it can be assumed that the feed velocity is much less than the tangential cutting velocity, and as seen from figure 2.2, the angle between the vectors is approximately zero, therefore it can be assumed that [3]:

$$V_w \cong V = \pi ND \qquad [2.2]$$

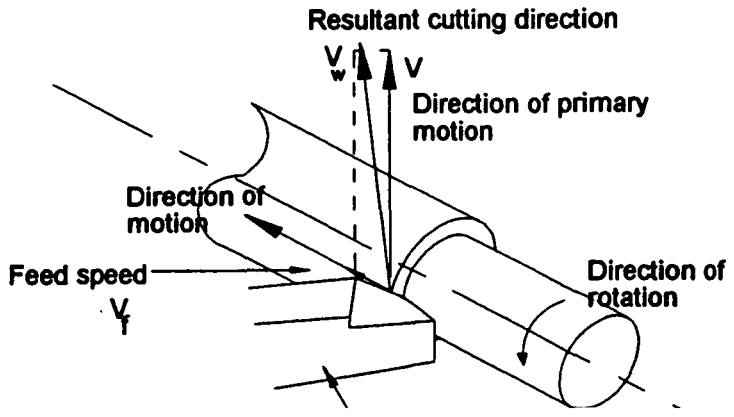

Figure 2.2 Cutting speeds of the turning operation [3]

Speed, always refers to the spindle and the workpiece. When it is stated in revolutions per minute (rpm) it tells their rotating speed. But the important variable for a particular turning operation is the feed velocity, or the speed at which the work piece material is moving past the cutting tool. It is simply the product of the rotating speed times the circumference of the workpiece before the cut is started. Every different diameter on a workpiece will have a different cutting speed, even though the rotating speed remains the same.

Feed, always refers to the cutting tool, and it is the rate at which the tool advances along its cutting path. On most power-fed lathes, the feed rate is directly related to the spindle speed and is expressed in mm (of tool advance) per revolution (of the spindle), or mmpr.

Depth of Cut, is the thickness of the layer being removed from the workpiece or the distance from the uncut surface of the work to the cut surface, expressed in mm. It is important to note though, that the diameter of the workpiece is reduced by two times the depth of cut because this layer is being removed from both sides of the work.

Given the above system parameters a model can be set up for these components from figure 2.3.

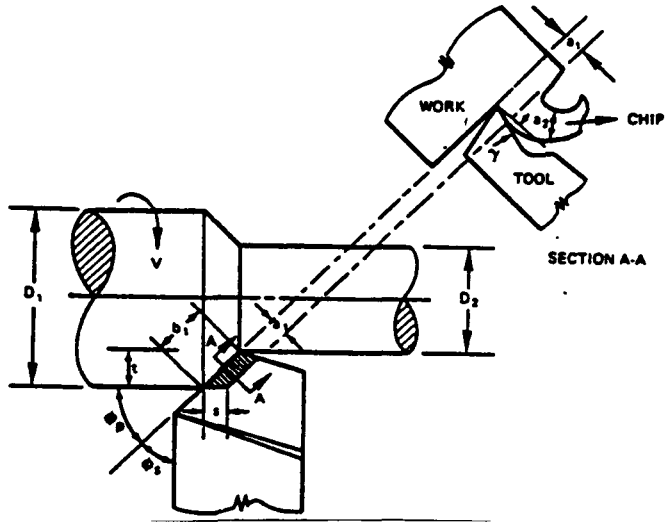

Figure 2.3 Cutting mechanics of a turning operation [2]

$$\textit{Cutting velocity}, V = \frac{\pi DN}{12} \qquad \textit{Depth of cut}, t = \frac{D_1 - D_2}{2}$$

$$\textit{Feed rate}, s_m = sN \qquad \textit{Principal cutting edge angle}, \phi_p = 90 - \phi_s$$

$$\textit{Feed} = s \qquad \textit{True feed}, a_1 = s\sin\phi_p = s\cos\phi_s$$

$$\textit{Time of cut}, T_c = \frac{L}{sN} \qquad \textit{True depth of cut}, b_1 = \frac{t}{\sin\phi_p} = \frac{t}{\cos\phi_s}$$

$$\textit{Material removal rate}, Q = 12stV \qquad \textit{Chip thickness} = a_2$$

$$\textit{Length of cut} = L$$

This theory describes the basic orthogonal approach to turning geometry. Furthermore the next section continues to outline the orthogonal cutting forces involved in the turning process. This is of particular importance as these two force components become an input to the neural network model for quantitatively predicting the surface finish online.

2.1.2 Force Relationships

Given the previous cutting conditions, the forces for traditional orthogonal cutting in metal turning can be analysed. It is important to consider the force requirements of the system, as these performance measures affect the selection of machine tools, cutting tools, optimal cutting conditions and most importantly the level of the surface finish quality acquired.

For a conventional single point turning tool the force system can be divided into three components:

1. F_{feed} = Feed force in the direction of tool travel. Controlling the axial thrust force on the spindle, and forces and power on the feed mechanism.
2. F_{radial} = radial force. Controlling work piece deflections during the cutting process, dimensional accuracy, and vibrational stability of the turning operation.
3. $F_{\tan gental}$ = Tangential force. Controlling the torque and power requirements of the main spindle, and the work piece and the tool shank size and cutting edge required to sustain the cut.

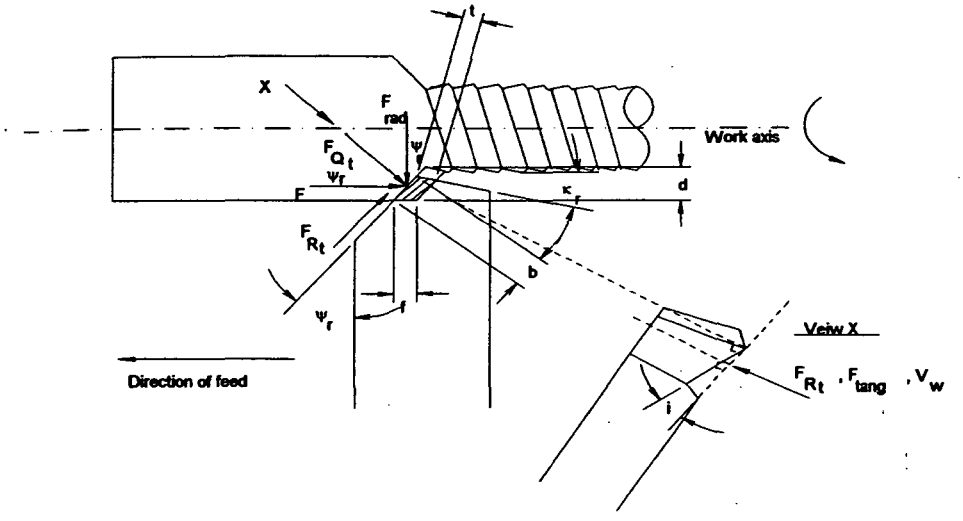

Figure 2.6 The cutting forces in reference to the cutting tool [3]

To analyse the cutting forces the application study of Dr Armarego [3] is to be used. Other simpler empirical equations have been used, but they relate only basic parameters (such as depth of cut and feed only), and rely on set work material and set geometry.

In reference to figure 2.6 we have:

$$b = \frac{b}{\cos(\varphi_r)} \qquad [2.3]$$

and

$$t = f\cos(\varphi_r) \qquad [2.4]$$

also

$$A \cong df = \frac{(b\cos(\varphi_r t))}{\cos(\varphi_r)} = bt \qquad [2.5]$$

ie we have a direct comparison between classical oblique and orthogonal cutting processes. The turning operation alters the equations in that:

$$V_f << V, d << D \text{ and } f << d. \qquad [2.6]$$

Given the previous diagram, the force components can be related to the classical cutting methods by:

$$\begin{aligned} F_{tangental} &= F_{P_T} \\ F_{feed} &= F_{Q_T}\cos(\varphi_r) + F_{R_T}\sin(\varphi_r) \\ F_{radial} &= F_{Q_T}\sin(\varphi_r) - F_{R_T}\cos(\varphi_r) \end{aligned} \qquad [2.7]$$

While these equations analyse the simple geometry of the turning process, they don't consider edge forces on the tool and work piece interface. Taking into account edge forces the equations become:

$$F_{\tan gential} = [K_{1P}b + K_{2P}A] = \left[\left(\frac{K_{1P}}{\cos\varphi_r}\right) + K_{2P}f\right]d$$

$$F_{feed} = \left[\left[\left(\frac{K_{1Q}}{\cos\varphi_r}\right) + K_{2Q}f\right]\cos\varphi_r + \left[\left(\frac{K_{1R}}{\cos\varphi_r}\right) + K_{2R}f\right]\sin\varphi_r\right]d \qquad [2.8]$$

$$F_{radial} = \left[\left[\left(\frac{K_{1Q}}{\cos\varphi_r}\right) + K_{2Q}f\right]\sin\varphi_r - \left[\left(\frac{K_{1R}}{\cos\varphi_r}\right) + K_{2R}f\right]\cos\varphi_r\right]d$$

where :

$$K_{2P} = \left[\frac{\tau}{\sin\phi_n}\right]\frac{\cos(\beta_n - \alpha_n) + \tan i \tan\eta_c \sin\beta_n}{\sqrt{\cos^2(\phi_n + \beta_n - \alpha_n) + \tan^2\eta_c \sin^2\beta_n}}$$

$$K_{2Q} = \left[\frac{\tau}{\sin\phi_n \cos i}\right]\frac{\sin(\beta_n - \alpha_n)_n}{\sqrt{\cos^2(\phi_n + \beta_n - \alpha_n) + \tan^2\eta_c \sin^2\beta_n}} \qquad [2.9]$$

$$K_{2R} = \left[\frac{\tau}{\sin\phi_n}\right]\frac{\cos(\beta_n - \alpha_n)\tan i - \tan\eta_c \sin\beta_n}{\sqrt{\cos^2(\phi_n + \beta_n - \alpha_n) + \tan^2\eta_c \sin^2\beta_n}}$$

$K_{1P}, K_{1Q}, and\ K_{1R}$ *are the edge force coefficients.*

where

τ = *shear stress on the shear plane*

ϕ = *shear angle*

α = *tool rake angle*

t = *chip thickness*

b = *width of cut*

β = *friction angle*

V_w = *resul* $\tan t$ *cutting speed*

The edge force coefficients displayed above are usually displayed in data banks for the specific material.

To reduce the complexity of the force components above, we can use the following relations relating shear plane angle, ϕ_n, and the friction angle, β :

$$Tan(\phi_n + \beta_n) = \frac{\tan i \cos \alpha_n}{\tan \eta_c - \sin \alpha_n \tan i}$$
$$Tan(\phi_n) = \frac{r_1(\cos \eta_c / \cos i)\cos \alpha_n}{[1 - r_1(\cos \eta_c / \cos i)\sin \alpha_n]} \quad [2.10]$$

and from geometry we obtain:

$$Tan\beta_n = \tan \beta \cos \eta_c \quad [2.11]$$

With the above simplification we now a function of the three force components in a reduced form:

$$F_{\tan g}, F_{feed}, F_{radial} = fn(d, f, \varphi_r, \alpha_n, i, \tau, r_1, \beta, K_{1P}, K_{1Q}, K_{1R}) \quad [2.12]$$

While this work identifies tool/work piece interference and associated dynamic modeling, the surface finish of the work piece needs to be qualified as part of the model. From equation 2.12, it can be seen the orthogonal modeling process has far too many complex variables for an online determination to be drawn from it. Therefore the need for a less complex analytical method. The following section discusses the basic orthogonal turning process and its effect on surface finish.

2.1.3 Surface Finish

The quality of a machined part is characterised by the surface finish obtained through machining [4], and the physical and mechanical properties of the surface layer of the material cut. The surface roughness depends on the tool geometry during the interface and the material removal process. Therefore as the working tool deteriorates, so too does the surface quality of the work piece. In a realistic machining situation the surface roughness profile and quality depend upon [6]:

- The type of chip formation.
- The chip flow.
- The cutting fluid used.

- The motion of the machine tool.
- The vibration associated with the machine and tool.

Figure 2.4 shows the geometry of the surface after machining by the cutting tool:

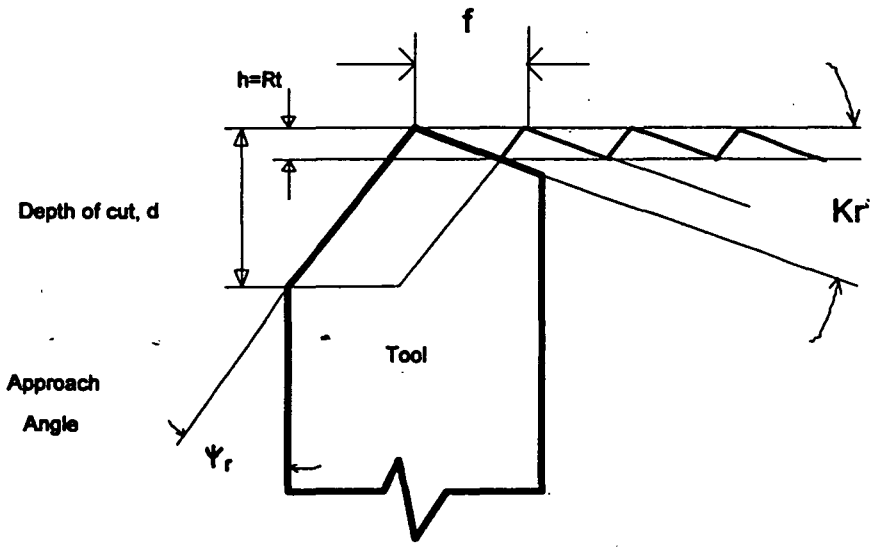

Figure 2.4 Surface geometry after cutting [3]

The surface roughness is characterised by two standard relations [3]. These are the peak-to-valley height, Rt (or Ry), and the centre-line-average, Ra.
From the geometry of the above diagram, the following expressions can be obtained:

$$R_t = h = \frac{f}{\tan\varphi_r + \cot\kappa_r} \qquad [2.13]$$

$$ie \quad R_t = fn(f, \varphi_r, \kappa_r)$$

Where [2.14]
φ_r = *approach angle*
κ_r = min *or cutting edge angle*
f = *feed*

(If the situation arises where Kr = 0, then an ideal surface will result.)

For the case of a simple tool, the surface roughness can also be determined by the centre line average, Ra. This assumes that the centre line is mid way between the peaks and valleys. Thus:

$$R_a = \frac{h}{4} = \frac{f}{4(\tan\varphi_r + \cot\kappa_r)} \qquad [2.15]$$

The equations above relate to a simple sharp edged tool. This is unrealistic as in practice a nose radius is given to the cutting tool. This nose radius occurs at the junction of the two flank planes and as a result the surface roughness is reduced. This is shown diagrammatically below:

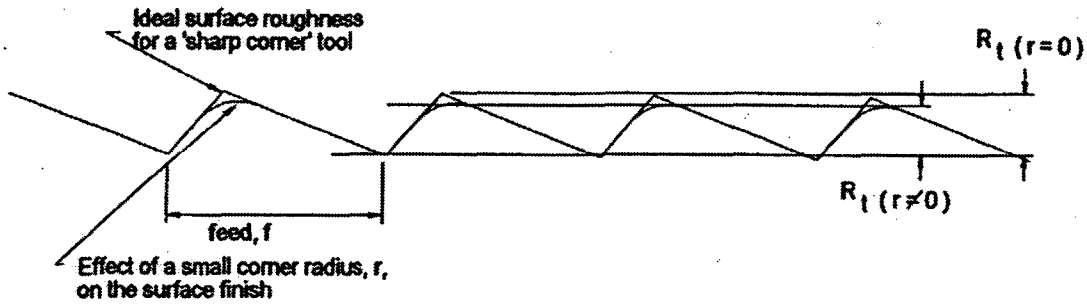

Figure 2.5 Effect of tool nose radius on surface finish [3]

Now, during the finishing process the feed rate is very small, and the feed marks are only caused by the corner radius.

$f < 2r\sin\kappa_r$
$d > h$

therefore [2.16]

$$R_t = (1-\cos\kappa_r)r + f\sin\kappa_r\cos\kappa_r - \sqrt{2fr\sin^3\kappa_r - f^2}\sin^4\kappa_r$$

An approximation to this formula has been given by:

$$R_t = h = r - \frac{1}{2}\sqrt{4r^2 - f^2}$$
$$R_t = h \cong \frac{f^2}{8r} \qquad [2.17]$$

similarly the center line average can be represented by the approximation:

$$R_a \cong \frac{f^2}{18r\sqrt{3}} \qquad [2.18]$$

The above sections summarise the performance characteristics for turning, and the result on the surface finish of a work piece material. The next section is directly related to the surface finish, in that, as the cutting tool deteriorates, the surface finish quality also decays at a proportional rate.

2.1.4 Failure of Cutting Tools

The cutting tool system and parameters have been set up in the notes above. Based on that information the analysis of cutting tool wear can be done. As materials are machined by a cutting tool, the tool wears and eventually fails. The rate, causes, and types of tool wear, and eventually failure, depends upon factors such as cutting conditions, workpiece material, tool material, and tool geometry. A cutting tool can fail by means of one, or a combination of the following processes [3,4,5]:

a) Gradual wear at the tool flank (flank wear) and/or at the tool face (cratering).
b) Mechanical breakage (ie chipping and microcracking).
c) Process of plastic deformation.

In assuming that the cutting edge has form stability under operating conditions (overcoming high temperature and stresses), then the main process of tool failure are, due to interactions between the chip and the tool (cratering) and between the tool and the work piece (flank wear).

Primarily tool failure under normal cutting conditions is usually gradual wear. The useful life of a tool (known as "tool life") is limited by tool wear. The wear can be described as the total loss of weight or mass accompanying the friction of the sliding or rotating parts. The wear is caused by five basic mechanisms:

1. Abrasion wear. Abrasion wear is caused by hard constituents of the work piece material, including fragments of built up edge, plowing into the tool surfaces as they sweep over the tool.
2. Adhesion wear. Where two surfaces are brought into intimate contact under loads and subjected to friction.
3. Diffusion wear. Where atoms in metallic crystal within the tool, shift from one lattice point to another causing a transfer of the element in the direction of the concentration gradient. The amount of diffusion is dependant on the period at which a high enough temperature is existent.
4. Chemical and electrolytic wear. Chemical wear is caused by interaction between the tool and workpiece in a chemically active cutting fluid environment. Electrolytic wear is caused by possible galvanic corrosion between the tool and the workpiece.
5. Oxidation wear. Oxidation also causes tool wear at high cutting speeds (in a high cutting temperature range). Through oxidation the tool cutting matrix is weakened, which in turn weakens the strength of the cutting edge.

The tool wear progresses as a cutting operation progresses. The wear land extends from the cutting edge up the flank of the tool. In addition, a characteristic cavity, known as a "crater", forms at a certain distance from the cutting edge on the tool face. The most common types of wear, flank and crater wear can be seen in the figure 2.7.

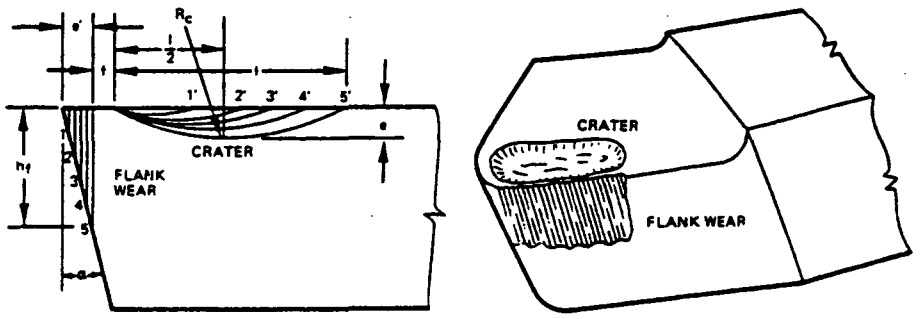

Figure 2.7 Geometry of crater and flank wear [2]

where:

h_f = *wear land*

l = *crater width*

e = *depth* [2.19]

R_c = *radius*

f = *distance from the cutting edge*

A typical representation of the development of wear with respect to time is shown in the following graphs:

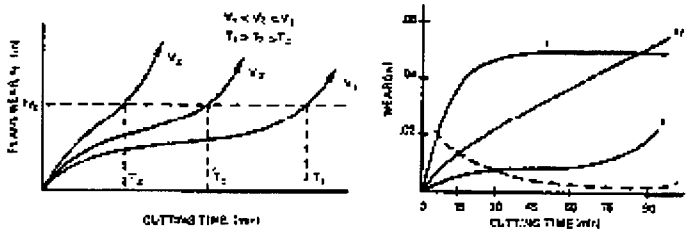

Figure 2.8 Relation of wear to cutting time [2]

Figure 2.8 demonstrates how the tool wear decays is dependent on the operating conditions. The decrease in surface finish quality is directly proportional to the rate in which the tool degrades. A wearing or worn tool will be evident through a higher frequency of vibration, and erratic force components. This will cause the peaks in the surface finish to be larger and less uniform.

The next section discusses how various methods and experimentation has been carried out to examine the gradual decrease in surface finish quality. These techniques have been performed offline, so the actual experimental conditions are not fully reflected. Online techniques will be discussed later.

2.2 SURFACE FINISH QUALITY MONOTORING

In production engineering the cutting process gets more significant in comparison to abrasive processes when machining hardened components. The application of this new technique shows a higher potential, because of the request for better production quality and productivity concomitant to lowering the production costs, abbreviation of the production sequence and minimization of processing times. The need for increased cutting life of a machine tool, and greater quality of the work materials surface finish, has led to the need for greater control over the material removal process. To gain this control, techniques have been developed to analyse various aspects of the cutting process, during or after cutting has been completed. Therefore the methods analysing the cutting process can be split into two main categories, online or offline.

2.3 OFFLINE MONOTORING

Offline techniques for measurement of tool wear and surface finish, involve examining machining parameters and performance, after the actual material removal rate process has occurred. Thus it is trying to equate a relation for the wear of the tool based on post machining data or trends.

Offline techniques are certainly a lot easier to implement, but it does not capture a realistic image of the actual system parameters. What this means is that the process parameters such as temperature, vibrations, forces and edge wear, cannot be modeled accurately offline. This is because these attributes return to their neutral state as soon as the tool leaves the work material. While offline methods may be inaccurate in their assessment of the above process parameters, they can give important information on such aspects as:

1. Reduction of tool area
2. Future predictions and trends
3. Surface finish characteristics

Various projects have been undertaken to examine tool wear and surface profiles via offline techniques, which can be summarised by the following approaches:

1. Numerical or statistical analysis
2. Microscopy
3. Geometric modeling
4. Force relations
5. Wear maps

These will be individually discussed in depth to examine their effective prediction capabilities.

2.3.1 Numerical or Statistical Analysis

Because it is desirable to replace a tool before it completely fails numerical methods have been developed to obtain a level of permissible tool wear. These are probably the most basic and dated techniques to determine the life of a tool or the expected surface profile. The first of these is the analysis of the Taylor tool life equation [2]:

The wear criterion for cratering is given by:

$$h_c = \frac{e}{\frac{1}{2} + f} \qquad [2.20]$$

which has common values of .4 for carbide, and .6 for HSS. Similarly values for the wearland length h_f for flank wear, are typically given as 0.025in - 0.03in for commonly used carbide tools. Taylor has proposed the relationship between cutting speed, V, and tool life, T, in minutes:

$$VT^n = C$$

where:

n = *Exponent depending upon tool material, workpiece material, cutting conditions, and environment.*

C = *Constant depending upon tool material, workpiece material, cutting conditions, and environment.*

[2.21]

This formula is able to obtain basic data, or tool life expectancies, for different varieties of cutting tool. This is illustrated in figure 2.9.

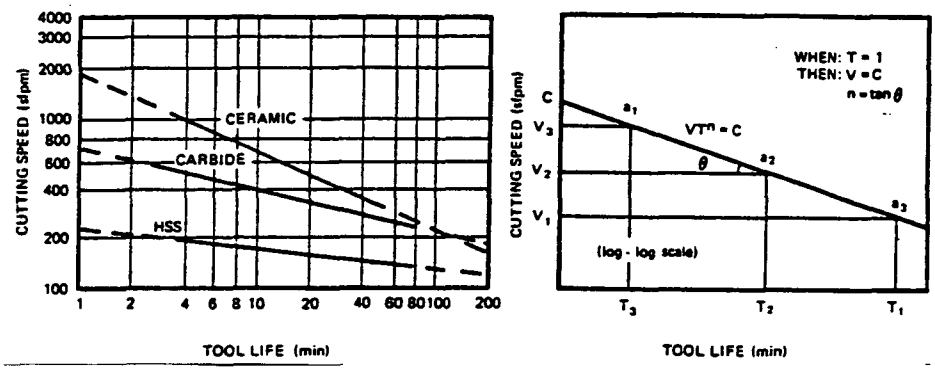

Figure 2.9 Tool life with response to cutting speed [2]

While the above equation (Taylor's tool life equation), only takes into account cutting speed and tool life, the equation can be expanded to incorporate feed, and the depth of cut required [2]:

$$VT^n s^\alpha t^\beta = K$$

where:

s = *feed*

t = *depth of cut*

α, β = *exponents for feed and depth of cut.*

K = *constant*

[2.22]

Taylor's tool life equation is still widely used as it gives a reasonable approximation of the tool life, but it only considers the above parameters, and it gives no meaningful relationship between the tool wear, and the conforming product quality output.

A more representative approach to offline management is Statistical Quality Control (SQC) [7]. SQC uses numerical techniques to analyse actual machining trends and can readily be used within the integration of online techniques. Statistical analysis can be put to good use in determining machine performance, especially indicating the near end of useful life of tools. By using SQC it introduces the possibility of having intelligent industrial machines to alert users of impending end of life of the machining tool. The statistical usage of SPC charts are explored for such purpose, and finding out how the recommended approaches be embedded into machines of the future.

Much has been said on the significance of SQC analysis on processes, detecting process variations and confining them to the control limits and specification limits. It has been attempted to apply the same principles and analytical power of SPC to the machining tools to highlight statistical features of impending tool wear and breakage.

In the same manner SQC helps in highlighting out-of-control process variations, the same philosophy can be applied to highlight successive out-of-process variations in the process. Making an effort to detect the small shifts in the process, and extending this over a number of successive chart points may indicate abnormal tool wear - which is a "warning" sign to abrupt tool breakage. Thus, this approach of using statistical analysis on tools is low-cost to implement (almost negligible cost), and can help to propagate the use of on-line SQC analysis in a manufacturing company.

Special charts are used for tool wear in statistical analysis, being cumulative sum (CUSUM) and exponentially weighted moving average (EWMA) charts. These charts, have several advantages over the traditional Shewhart (x-bar) chart, in that they are more sensitive to smaller process shifts, less likely to signal false alarms, and are relatively immune to the effects of measurement noise.

An EWMA (Exponentially Weighted Moving Average) chart is used when it is desirable to detect out-of-control situations very quickly. EWMA charts have a built in mechanism for incorporating information from all previous subgroups (data points), weighting the information from the closest subgroup with a higher weight.

Thus, the control/out-of-control decision is made with information from previous subgroups as well as the current subgroup. The chief advantage of EWMA charts is that they detect out-of-control conditions more quickly than X-BAR charts and that this detection can be done by using only one rule, that is, being within or outside 3 sigma (sigma is standard deviation) control limits.

A gradually wearing-out tool produces small shifts in the process, which are unlikely to be detectable using the X-BAR charts. The EWMA chart is a good choice because of its intrinsic property of assigning a heavier weight to earlier data points, and gradually a lighter weight to later data points. The direct result will instill a geometrically "ageing process property" to the data points, which is exactly what we is desired to emulate in an ageing (wearing-out) tool. Changing the lambda to smaller values would cause the EWMA chart to detect smaller process shifts and vice-versa.

Judging from this inexpensive approach to tool wear detection, process sensors can be installed on CNC machines, which serve two purposes synergistically and concurrently, being the implementation of normal SPC on processes and automating tool wear detection indirectly.

While this gives a reasonable prediction there are more accurate measures of tool wear, for example, using chip-embedded tool tips, image processing and neural networks to detect patterns of impending tool failure. These techniques may be more costly to implement but their predictive capabilities are far superior.

Yellowley [8] used a similar analytical approach, in examining the force ratios of a turning system, and its impact on the life of a cutting tool. The use of force information, and in particular the ratio of thrust force to power consuming force, is used to analyse both tool wear and tool breakage in turning. The work by Yellowley [8] addresses the issue of the influence of machining conditions in the turning of typical industrial work pieces. Such work pieces contain tapers, arcs and other elements, over which the geometry of cut may influence the force ratio extensively. The influence of cutting conditions on the force ratio has been pointed out in previous works, however, these investigations have suggested that this was not a

critical problem. The work in this offline study reveals that in a more realistic turning application, the force ratios are impacted by the wear of the cutting tool.

In a very similar offline approach S. Elanayar, and Y. C. Shin [9] examine the relation for the separation of ploughing forces from shearing forces on the shear plane, and its corresponding impact on the work piece material and tool piece. The forces are decomposed by first separating the shear forces from the total forces and then employing an iterative procedure to calculate the normal forces on the shear plane. The procedure uses the indentation models along with values of tool and work piece material constants to determine the indentation force. Models for the indentation depth have been developed from the designed experiments and the predictions by the established models were compared with experimental results obtained for different cutting conditions.

Once again as with the above technique, the work gives an approximation at an online situation, and hence rough guidelines for the wear rates and surface finish of the work material. However the process is too restricted in that it only considers very few machining parameters, and offers no concrete solution to accurate monitoring.

While these methods give some insight to the problem, they are a mainly a "text book" or empirical approach, which offers very little practical evidence of a solution. Microscopy, while still very limited, gives some practical evidence towards the tool wear, surface degradation problem.

2.3.2 Microscopy

On the other hand microscopy is very common in analysing the wear zones of a failed tool. It is very much an offline measurement, as cutting fluids, vibrations, chip formation, and tool/work piece interaction, make it virtually impossible for online vision. Through the use of a microscope the wear regions of the tool (crater, edge etc) can be obtained, so that an estimation can be made to the useful life of the tool in the given situation, based on the worn tool.

G.M Zhang, T.W. Hwang [10] have presented a new approach to the study of cutting dynamics in microscale. The manipulation of the cutting force generated during machining is based on the characteristics of microstructures within the work piece material being machined. Mathematical modeling of the hardness variation around the circumference of the work piece reveals the cohesiveness between the macroscale and microscale analyses. The case study illustrates the procedures used to evaluate the cutting force through the microscale analysis. A model-based indirect tool wear monitoring methodology has been developed to show the potential of applying the cutting dynamics in microscale for the design of on-line quality and process control systems.

While microscopy gives some experimental evidence into the decay cycle of the tool and the corresponding surface finish, there are not enough process variables exhibited to make it a feasible process prediction method. Also it is offline, and not a feasible process to implement online, therefore a reliable prediction cannot be achieved for a range of machining conditions. A similar restrictive approach is geometric modeling.

2.3.3 Geometric Modeling

Geometric modeling is an almost identical process to microscopy, in that its desired output variables are the same. Different geometrical methods are used however to obtain the output.

When machining hard (high strength) steels with geometrically defined cutting tools, the geometry of the cutting tool changes continuously due to various wear mechanisms and therefore variations in component quality must be expected. Various study has been carried out to investigate the fundamental relationships which exist between process parameters, cutting tool wear and the resulting surface integrity when machining hardened steels. The complex geometry of worn cutting tools is quantified using specially developed image analysis software. Apart from the usual surface roughness descriptors, surfaces are examined for structural changes arising from the high temperatures and stresses developed in the chip formation zone.

S. G. Kapoor and R. E. DeVor [11] have taken a geometrical approach to the analysis of a worn cutting tool and its impending effect on the surface quality, but have also based it on force modeling as in the first section. The effects of tool wear, on the economic viability of the machining process is immense. A dramatic rise in cutting forces is typically associated with a worn tool flank. This has a serious effect on the tool/work piece deflections which result in dimensional inaccuracy in the finished part. The geometric change in the tool itself resulting from flank wear causes undesirable changes in the dimensions of the cut. The high temperatures and stresses which result from the increased friction and rubbing of a worn tool also strongly affect surface quality, cutting power, and stability.

A force model that simulates the effects of wear and predicts the resulting changes in forces accompanying a worn tool has the potential to alter the current approach to wear and greatly improve process efficiency and tool utilization. Not only can such a model be used during design to estimate part errors, fixture requirements, and tool life, but it can also be incorporated into an on-line tool condition monitoring system for estimating current wear, scheduling efficient tool changes, and detecting the deterioration of part quality based on a measured force signal.

The proposal of S. G. Kapoor and R. E. DeVor describes an in-depth study of tool wear aimed at developing a wear-force relationship and modifying existing mechanistic modeling approaches to incorporate that relationship. Their focus is on flank wear and the prediction of cutting forces given a specific geometry of work tool flank. The reverse problem is also to be considered in which the current flank geometry is to be estimated given the force signal. The model has initially been developed for orthogonal cutting and then, as was accomplished successfully in sharp-tool models, the approach is to be extended to the traditional three-dimensional processes described above. Specific objectives are as follows:

1. Develop an analytical relationship between the fundamental parameters governing the orthogonal cutting process (including flank wear geometry), such as material shear flow stress, shear angle, friction coefficients, and tool

geometry and the basic force system defined on the shear plane and flank wear land.

2. Apply the mechanistic modeling approach to the analytical orthogonal cutting model to relate the measured forces on the tool to the traditional process inputs (i.e., feed rate, cutting speed, flank geometry, etc.) in such a way that expensive, time-consuming wear experiments are not needed for calibration.

3. Extend the mechanistic model for orthogonal cutting of worn tools to the three-dimensional sharp-tool models which exist for face/end milling, boring, an drilling.

The work achieved by the offline technique, gives Industrial users of metal-cutting tools an empirical answer to the problem of tool wear. The proposed research enhances existing process models to include the effects of wear. These models, now well established for predicting process outputs from a sharp tool, will be modified to accept a level of flank wear as an input to the process. The resulting prediction capability will allow designers to understand the quantitative effects of wearing a tool to a certain level, thus aiding in the selection of processes, fixtures, and product dimensions. Thus product and process design can occur simultaneously, with consideration given not only to the first item produced with a perfectly sharp tool by also to the many items produced before a tool finally reaches the end of its useful life. The model can also be used to better establish a criterion for tool life as well as monitor the process to detect when that life is near its end.

While this process delivers a model for tool life, it relies on the assumption that the tool life is reliant only on the flank wear of the tool, based on a range of tested tools. Thus in fact, while this process gives an approximation to tool life, users relying on the information could be removing tools that are still very much conforming to company standards. In general, even though it is a reasonably inexpensive offline process, it is completely unfeasible to put an approximated solution online that governs so few machining parameters.

C Y H Lim [12], has used a geometrical approach, in that of wear maps. A wear map is a multi- (usually two) dimensional graphical presentation of wear data. It is a similar approach to geometric modeling or microscopy, where the wear of the tool is monitored offline. However unlike most other conventional approaches, which tend to restrict the breadth and depth of wear information reported, a wear map is able to show a fuller picture of a wearing system by linking measured wear rates with observed wear mechanisms over a much wider range of conditions. Wear maps have been developed successfully for the sliding wear of metals, ceramics, metal-matrix composites and polymers, and for the wear of uncoated tools in actual machining operations.

The current project investigates the wear behavior of titanium carbide- (TiC-) coated cemented carbide tools in turning operations. Experimental tool wear data from carefully executed single point dry turning tests, together with similar data culled from the technical literature, are used in the construction of wear maps showing the flank and crater wear characteristics of these tools over a wide range of machining conditions. The maps are shown here in Figures 2.10 and 2.11, respectively. The axes of the wear maps are defined by cutting speed and feed rate, which are the two most important factors that affect tool wear. The boundaries on the maps demarcate areas of machining conditions within which the same ranges of wear rates are observed.

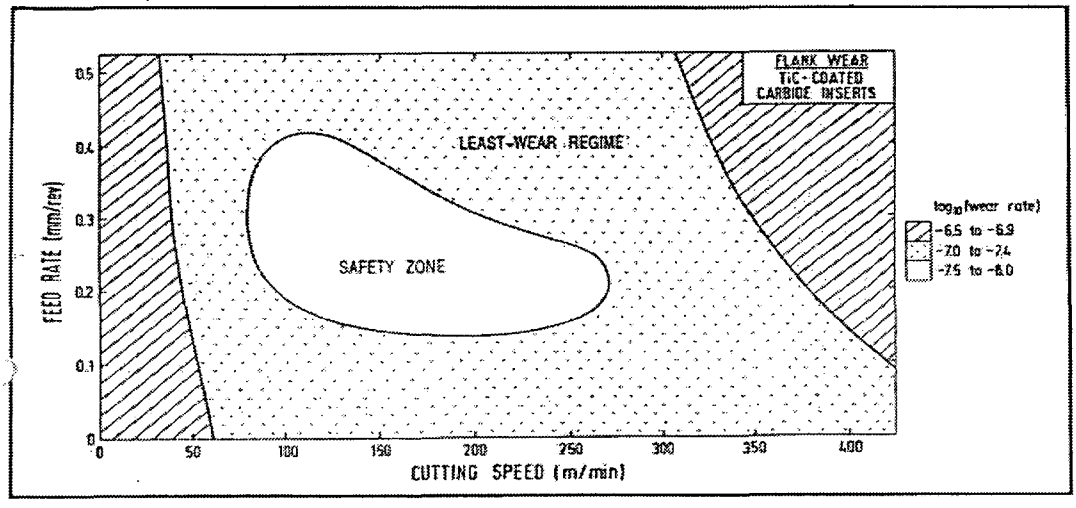

Figure 2.10 Wear map showing flank wear characteristics [12]

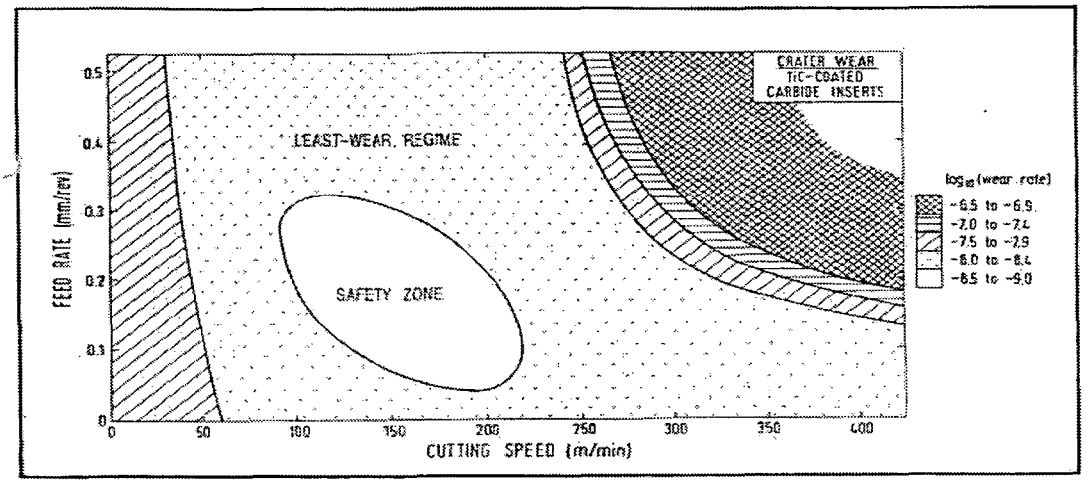

Figure 2.11 Wear map showing crater wear characteristics [12]

Both the maps show that the extent of flank and crater wear varies according to cutting speed and feed rate. On each map, there exists a range of conditions, denoted the "safety-zone", under which the coated tools would suffer the smallest amount of flank or crater wear. Also of significance are the regions of the next lowest wear rates (shaded with dots). These cover far wider ranges of machining conditions than the safety zones and allow greater flexibility in the choice of cutting conditions that will still result in a low wear rate.

While this method offers a reference for machining, it offers only a guide to turning tool wear. There is no relation found between the wear maps, and the surface finish profile. Also rather than decreasing the quality control the quality control period, it ineffectively increases it. The importance to the customer is the quality of the product they are receiving, therefore the emphasis must be placed on the quality of the product, as opposed to the wear on the machine.

While offline methods of surface finish and detection still offer some benefits, and are often the only feasible possibility, there is a much greater demand in today's industry for tighter manufacturing control. The following section, discusses current principles, or attempts towards methods for insuring a completely online and automated production line.

2.4 ONLINE MONOTORING

One of the most important manufacturing process is machining. According to some researchers, it turns into chip nearly 10% of the global metal production. Moreover, machining is responsible for more than 40% of the production cost of different types of parts [13,14,15].

In recent years machining processes have suffered dramatic changes aiming to increase productivity and quality assurance, which has been reached by means of process automation, e.g., CNC machines. However, in order to have a fully automated machining process, it's necessary to use online tool life monitoring systems, which will lead to establishing a quantitative surface finish prediction. Nowadays the strategies used, to establish the region of surface wear are based on statistical data. But due to machining process complexity and unpredictability, the surface finish range is very wide. Therefore, many tools are replaced in conditions to be used, which increases tool consumption and dead time. For the reasons described above the use of monitoring systems may be very useful to optimise tool life and decrease production costs.

Online or 'in-process' surface finish monitoring has attracted wide interest for a long time. It has presently acquired more importance than ever as manufacturing systems are increasingly required to provide greater automation and flexibility, while keeping high productivity levels. In unattended manufacturing, worn tools must be changed on a statistical basis, eg., with reference to the shortest life expectancy of the tools used. Such procedures are inefficient as much of the useful life of the tools may be wasted and high tool changes would mean higher machine downtime, decreasing thereby the system productivity. In-process monitoring techniques which allow for the optimal utilization of tool life, though highly desirable, are to date not yet fully developed.

Various fields of study and investigation have been carried out, on an online basis, to determine a direct method of cutting tool wear detection. These fields of research can be categorized as:

1. Acoustic Emission
2. Vibration/Damping
3. Sensor Fusion
4. Neural Networks
5. Optical computer vision
6. Metallurgy
7. Radiography
8. Fuzzy Logic
9. Force modeling

And each previous investigation or field of study, may recognise or utilise one or more of these online techniques.

2.4.1 Acoustic Emission

Acoustic emission study, relates the noise frequencies from the machining process, to the wear region of the cutting tool, and also the corresponding surface finish level. Different wear types (wear of tool flank face and tool chipping) result in changes in the different characteristic values of the noise signal. In case of a uniform abrasion of the insert, e.g., flank face or crater wear, an increased mean signal level is observed, whereas for microbreakage at the edge, an increase of the crest factor with nearly constant mean signal level is found [13].

S. Cho, and K. Komvopoulos [16] have studied the correlation between Acoustic Emission and Wear of Multi-layer Ceramic Coated Carbide Tools. Acoustic Emission (AE) was used to monitor the machining process and tool condition during turning of AISI 4340 steel with uncoated, two- layer TiC/Al2O3) coated, and three layer (TiC/Al2O3/TiN) coated cemented WC-Co tools. The experiments were performed at four different feed rates and constant cutting speed and depth of cut. The variation of the AE signal with cutting time is interpreted in light of the dominant mechanisms, rates, and patterns of wear and the contact friction conditions at the tool/work piece and tool/chip interfaces.

It was shown that AE frequencies in the range of 50 to 100 kHz are primarily due to plastic deformation in the near-surface tool regions and the primary, secondary, and tertiary shear zones of the work piece. Cracking leading to coating delamination and WC grain pull-out generates frequencies in the range of 170 to 200 kHz.

This study generates positive feedback, relating values of acoustic emissions to the region of tool wear and level of surface finish. The main problem is that the online study gives no further progress into the online detection of surface finish quality, especially since the machining parameters are mostly kept constant, the AE data gives no true indication for a broad spectrum machining process.

J. A. Rice, S. M. Wu, Reid and Polly Anderson [17] have compiled a similar study on the feasibility of catastrophic cutting tool fracture prediction via acoustic emission analysis. The prediction of catastrophic cutting tool fracture and is effect on the surface finish is explored through monitoring the acoustic emission (AE) from a cutting process. A prediction parameter is derived which combines the AE signal with a physical model of a cracked tool to form an estimate of the spatial energy release rate.

Monitoring the energy release rate is found to be largely dependent on the detection of crack advancement. Experiments were performed with both new and artificially cracked inserts during interrupted cutting. Epoches denoting crack advancement were detected through high time homomorphic analysis of the acquired AE signals. AE bursts prior to and leading up to fracture were analyzed for crack advancement. The calculated energy release rate was found to exponentially increase as fracture was approached. Crack advancement could be feasibly detected approximately six cuts prior to fracture.

While this study advances on the last, once again the data is coherent only to the cutting conditions of their individual setup. Also it is completely unfeasible to integrate an online based knowledge into the cutting routine, when it is so restricted by cutting variables, and its accuracy questionable. The surface finish quality can only be categorised into approximate regions, by the determination of the region of

the tool wear. The data is too unrestricted to pinpoint a known region of surface quality.

2.4.2 Vibration / Damping

The study of vibration and damping involves analysing frequency of vibration of the cutting tool or its directly related components. From the frequency output, parameters can be established to the region of the tool wear and the corresponding surface finish quality.

A.A. Hood [18] has designed a control system for Active Vibration Control of a turning process using PMN actuators. In recent years, intensive research has been conducted concerning the use of smart materials for active vibration control and also vibration attenuation for machine tools. This research investigates vibration control of a machining operation using actuators made of smart materials.

In the study, the effectiveness of using electrostrictive actuators for active vibration control during a turning process on a conventional engine lathe is investigated. The actuators are made of Lead Magnesium Niobate (PMN) and are in a multi-layered configuration. The test bed is a steel structure, called the smart tool post. It is designed to be a key component of a conventional engine lathe machine tool and its purpose is to transmit compensating energy from the actuators to the tool tip during machining. The unique characteristics of PMN actuators to provide accurate displacements ensure the performance of vibration cancellation on the micron scale. The focus of the research was the design of a control system for performance optimization. The research combines both analytical and experimental approaches. In the analytical aspect, optimal and adaptive control schemes are proposed. Mathematical models of the smart tool post, based on first principles, are derived and evaluated. In the experimental aspect, system identification of the tool post's dynamics as well as the actuator's dynamics is performed.

Results from this thesis show that PMN actuators are good smart material, candidates for active vibration control. Optimal, and adaptive control, designs are critical to achieve effective broad band vibration compensation. This thesis gives a systematic presentation of the smart tool post's system modeling, control system design, and real-time microprocessor implementation. But most of all, the study illustrates that the use of PMN ceramic material as actuators does indeed have practical applications in precision machining.

Damping capacity greatly affects the dynamic rigidity of a machine tool system. The thesis paper by M. Hashimoto, S. Kato [19], deals with the damping capacity of turning tools. Damping capacity is measured as a function of clamping conditions such as clamping load or surface topography of tool shank. An empirical equation of damping capacity representing the clamping load dependence is induced for the fundamental mode of vibration in tangential and normal directions. The optimum clamping load, at which the stability of turning tool for chatter excitation becomes a maximum, is experimentally clarified.

A second field of study was attempted [20]. This was to examine the mechanism initiating the damping capacity of a turning tool. The turning tool is considered as a beam on an elastic foundation. Parameters of the model are estimated from the experimental result of the free damped vibration frequency. Next, the magnitude of damping energy is calculated from the friction resistance and the relative slip between the tool shank and the tool post (elastic foundation) during vibration in both tangential and normal directions. The calculated result of damping energy agrees well with the experimental result of the damping capacity qualitatively, for various clamping loads and various surface topographies. This result indicates that the damping capacity of the turning tool system is mainly caused by the relative slip at the tool shank.

The research above has been conducted mainly to determine the damping vibration that is inflicted on the cutting tool. There is no direct relation in their study to the wear of the tool or level of surface finish achieved. Research has been done to develop a system model which can describe the intermittent turning process more

precisely. In the development of the system model, the intermittent turning process is mathematically formulated and analyzed as a system associated with three periods, namely, the impact, cutting, and non-cutting periods. The main investigative methodologies used were:

1. Applying modern control theory to perform dynamic cutting analysis in the cutting and non-cutting periods.

2. Employing wave propagation theory to study the impact force during the impact period.

3. Using numerical simulation and finite element method to examine the impact process and its effect on the machining performance.

The emphasis is focused on the effect of impact on the tool motion and the effect of the work piece damping on the machining performance. The research results indicate that the impact between the cutting tool and the work piece at the every beginning of each cutting period is the main cause leading to mechanical failures of tool during the intermittent machining. Laboratory experiments have been conducted, and the obtained results show a general, good agreement with the results predicted by the developed system model. An attempt has been made to construct a stability boundary map.

This field of work is very practically unfeasible. While it has been discovered that the main tool wear through vibration occurs in the impact between the tool and work material, there is no conclusive evidence as to which point this will cause the tool to fail, or any prediction to which point in time that it may occur. And further more, there is no indication of the discreet relation between the vibration signals, and the state of the surface finish of the output component.

The work done by Fábio Nogueira Leão [15] attempts to improve on the previous attempts, which have been made on the development of monitoring systems. This study consists, basically, on relating the machining process parameters to the tool condition. During the process, friction between the chip/tool and tool/work piece

interfaces produces vibrations. Cutting force oscillations also produce vibrations. As cutting time elapses and tool wears, the frequency of vibration changes. On the other hand, electrical power and current consumed by the machine tool are proportional to the mechanical power consumed by the machining process and also proportional to cutting force. Therefore, a system that is able to relate variations in the vibration frequency or variation in the motor current to the tool wear may be useful to assure online tool life monitoring.

The main goal of this work is to correlate vibration signal variation and motor current variation with surface wear and the tool condition, when finish and rough turning. The experimental set up presented below. Experiments were carried out in order to compare vibration signal and current sensibility in the online monitoring system.

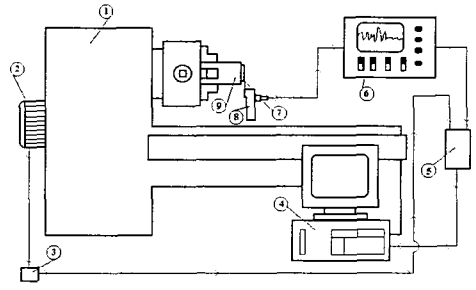

Figure 2.12 Experimental set up of Fábio Nogueira Leão [15]

The components of this experimental setup are listed below:

1- CNC power lathe
2- Main motor
3- Hall element
4- Computer
5- A/D board
6- Signal analyser
7- Piezoelectric accelerometer
8- Cutting tool
9- Work piece

This work approaches the goal of obtaining a quantitative surface finish approximation, in that two major process parameters are considered. The main problem is that there is still not enough experimental variables to be able to enable a smart system to be incorporated, and the testing variables limit the process to this particular system. There is very little variance of turning conditions allowed for. Furthermore the process is concerned with evaluating the tool wear, and offers very little reference to the quality of the produced components.

The predictive capability of this form of online system is quantitatively unreliable. A more reliable method of obtaining data is through the use of sensors.

2.4.3 Sensor Fusion

Sensor fusion simply utilises sensor components to analyse the wear of a tool or the corresponding surface finish quality. While study is concentrated on sensor fusion, the online models are usually associated with several other techniques, such as neural networks or fuzzy logic.

Kai Frank Goebel [21] has provided a means to deal with uncertainty in complex sensor driven systems, in particular for sensor validation, sensor fusion, and diagnosis. These means come from probability theory, neural network theory, and fuzzy logic and are more generally termed as "soft computing" techniques. The investigation considered systems that had to be controlled based on monitored sensor readings to allow corrective action in case of aberrations from a desired value. Since all sensor readings are always subject to a variety of noise conditions, such as Gaussian noise, bias, clutter, outliers, and non-symmetric noise distributions, the aim is to perform sensor validation and Ð in case of multiple sensor readings, ie sensor fusion.

This dissertation provides means for fuzzy sensor validation and fusion. The approach was compared with a probabilistic method, which is a Kalman filter based scheme assuming Gaussian noise distribution. While the fusion is performed for the probabilistic method in a Bayesian way, the fuzzy sensor validation and fusion approach uses non-symmetric validation regions in which sensors readings are

assigned confidence values. Each sensor has its own dynamic validation curve, which is shaped according to sensor characteristics. These characteristics can take into account the range, external factors affecting the sensor, reliability of the sensor, etc. The curves have its maximum value at the predicted value which is arrived at using fuzzy exponential weighted moving average time series predictor. Sensor readings have value 0 at the boundaries of the validation gate which is determined by the maximum possible change a system can undergo in one time sample. Since readings outside the gate are implausible, they are discarded. Readings closer towards the predicted value are rewarded with a higher confidence value.

Fusion is performed using a weighted average of sensor readings and confidence values and the predicted value scaled by the operating condition. Each method performs best in the presence of certain types of noise and recommendations are made as to which approach is more appropriate under various conditions. Another aspect of this dissertation is to provide a tool for diagnosis in the presence of vague symptoms. This is achieved though fuzzy abduction which can diagnose crisp as well as soft faults. This means that faults can be diagnosed if they occur to some degree. The proposed algorithm computes a closeness measure taking into account the distance from an observed symptom set to the modeled symptom set for all failure combinations. It then ranks the failure sets according to maximum closeness measure and minimum cover, i.e., number of faults. As an extension, a framework for fuzzy influence diagrams is provided which uses this closeness measure. Several applications from extant systems show the feasibility of the approaches developed here. From a manufacturing point of view, it shows how the neural network tools provided allow for improved decision making for optimal tool exchange in a high speed milling machine environment by monitoring the degree of component wear on-line. This process approaches a reliable prediction model, but with input data only consisting of ideal (non-discarded) values and ideal experimental conditions (conforming noise signals), evaluating the trends becomes unreliable when considering a real cutting scenario.

X. D. Fang, and Y. L. Yao [22,23], have studied In-process Evaluation of the Overall Machining Performance in turning, by the use of sensor fusion. Sensor fusion often uses multiple sensors to evaluate a single quantity. The work presented in this paper

attempts to use information from a single sensor to estimate overall machining performance (characterized by cutting forces, chip breakability, surface roughness, and dimensional deviation due to tool wear). In particular, the performance is aimed at reflecting the in-process change of the above-named quantities with respect to tool wear progression (major flank, crater and minor flank wear). 3D cutting force measured by a tool dynamometer is fully utilized by aggregating multivariate time series models and neural network techniques.

While the last field of study concentrates on the wear detection of the cutting tool, the following study by Y. C. Shin, S. J. Oh, S. A. Coker [24], concentrates on the analysis of the surface finish deterioration through sensor fusion.

The paper presents a new surface roughness measurement technique suitable for in-process monitoring in production machining processes. The technique is based on the utilization of focused ultrasonic beams with non-zero incidence angle and measurement of reflected intensities. Fundamental principles and the measurement system developed are described along with experimental results. The tests were conducted to test the sensitivity and robustness of the proposed technique with various machined surfaces and materials. The results show excellent correlation's with those obtained by a profilometer. In addition, it is shown that the periodicity's of the surface profile corresponding to feed marks can be extracted from the measured ultrasonic signals, thus providing the possibility of profiling. Therefore in-process monitoring of surface roughness during machining is demonstrated with the proposed technique, yet as suggested this method contains a restricted prediction, having little respect for any other cutting variables.

While the main focus in this thesis is on surface quality, the main influencing factor on the surface finish is the wear of the cutting tool. J. Colgan, H. Chin, S. R. Hayahi, and K. Danai [25], have studied a multi-sensor method to on-line tool breakage detection in the turning process. This incorporates a multi-sensor tool breakage detection system that characterises the state of measurements during normal (no-fault) condition and at tool breakage by the two columns of a multi-valued influence matrix (MVIM). In the system the measurements are monitored on-line and flagged upon the detection of abnormalities. Tool breakage detection is performed, by

matching the sector of flagged measurements against the two columns of MVIM, which are estimated during a training session so as to minimize the error in detection. The detection system is implemented in turning. Experimental results indicate that this system provides excellent detection when the full range of tool breakage effect on the measurements is included in training, and that its performance is less dependent upon the training set than a multi-layer neural network net. While beneficial, this method has very little variance in its training data, and while an approximate tool life can be associated, there is no reference to impending tool wear.

In another similar field of study [26] a three axis milling machine is being upgraded with an open architecture CNC control system. The machine forms a test-bed for research and development into sensors for monitoring part, machine, and process parameters. The machine is currently being modified, the servo loops being tuned, and the controller programmed for compatibility with the machine. The project aimed for:

1. Integration of Vision with CNC Machining to perform in-cycle measurements on parts using a 3D vision system.

2. Tool Wear Monitoring Via Image Processing Employment of image processing techniques to measure flank wear

3. Sensor Fusion to combine multi-sensor data from the test-bed and to utilise the result through the open architecture CNC controller for improved machining performance.

These aims have been proven to be illegitimate. In-cycle vision is far to restrictive to be quantitative, and image processing techniques are an offline method of detection. Therefore any employment of this data will result in unacceptable online prediction capabilities.

Many direct and indirect methods have been attempted to measure tool wear [27-31]. Recently, some interest has been expressed in utilising simple image processing

technology to measure flank wear on cutting tools. Unfortunately, these approaches have utilised simple binary images achieved through image thresholding. A field of research project is devising more sophisticated approaches using gray level images. More parameters can then be used to predict wear; for example, surface texture and wear land morphology. Previous research has not addressed the important issues of measurement repeatability and the advantages of one wear parameter relative to another. Only basic comparisons with a tool makers microscope have been performed.

The work mentioned above is almost an entirely research based study. No conclusive practical application is evident. The main problem with the study, is that the online implementation (through sensors and neural networks possibly), is based on the training information given by optical techniques. No image processing online technique has been found reliable enough on its own to continue further implementation. Image processing techniques can identify a general trend of tool wear for a very constant and ideal process, but in the machining industry there is a requirement for far more integration and variance.

Neural Networks are a form of smart system (Neural Networks are mentioned extensively in chapter 3). They are seen as the most effective form of artificial intelligence in the modern manufacturing industry. The following literature reveals the previous works done by researchers using neural network technology, to integrate the surface quality from an offline variable, to an online variable.

2.4.4 Neural Networks

In information technology, a neural network is a system of programs and data structures that approximates the operation of the human brain. A neural network usually involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially "trained" or fed large amounts of data. A program can then instruct the network how to compute in response to external data, or can initiate decision making on its own [32].

Neural networks often use principles of fuzzy logic to make determinations. Neural networks are sometimes described in terms of knowledge layers, with, in general, more complex networks having deeper layers. In feedforward systems, learned relationships about data can "feed forward" to higher layers of knowledge.

Tool failure and uncontrolled wear are the main reasons for poor quality in manufacturing of desirable components and expensive secondary costs. A decision making approach is therefore needed for "intelligent" manufacturing control. On-line surface wear and tool condition monitoring is desirable in an unattended manufacturing system since an unexpected interruption of the process for tool replacement due to excessive wear or fracture could halt production and affect the quality of the product. Neural networks are the most up to date technology as an intelligent online monitoring system.

Choon Seong Leem, D. A. Dornfeld, and S. E. Dreyfus [33] have worked on a basic customized neural network for sensor fusion of acoustic emission and force in online detection of tool wear. It is developed based on two critical concerns regarding practical and reliable tool-wear monitoring systems; the maximal utilization of "unsupervised" sensor data and the avoidance of off-line feature analysis. The neural network is trained by unsupervised Kohonen's Feature Map procedure followed by an Input Feature Scaling algorithm. After levels of tool Hear arc topologically ordered by Kohonen's Feature Map, input features of AE and force sensor signals are transformed via Input Feature Scaling so that the resulting decision boundaries of the neural network approximate those of error-minimizing Bayes classifier. In a machining experiment, the customized neural network achieved high accuracy rates in the classification of levels of tool wear. Also, the neural network shows several practical and reliable properties for the implementation of the monitoring system in manufacturing industries.

This approach gives inappropriate training to the neural network model in that it relies on the data only from acoustic emission and forces on the tool, to determine the wear zone of the tool. While tool may be worn, without knowing the corresponding surface quality output, it is impossible to justify the replacement of the

tool. This is the problem not yet achieved by this previous field of study. While they recognise that the study may be useful, the actual transition to online application is unfeasible without a reference variable.

The detailed work of D. Barschdorff and U. Femmer [34-44] in the study of artificial neural networks for wear estimation gives a slight improvement on the above testing. It consisted of two main works, similarly to this project. First was the experimentation, to obtain a knowledge base for the neural network. Secondly the neural model was incorporated into the process.

The testing process restricted the variables by keeping cooling and depth of cut constant, and varying cutting speed (160 to 200 m/min), and cutting feed (0.16 to 0.2mm). The main process variable being tested online was the frequency of vibration of the cutting tool. For featured selection only patterns determined from signals measured by using new and worn tools were used. The number of features could be reduced to five without losing classification accuracy. The selected features describe the power in single frequency intervals from 5-60kHz in relation to the power in the frequency interval 0.2-90kHz.

The figure below (figure 2.13) shows the calculated spectrum from the vibration signal created of a sharp and a worn out cutting edge.

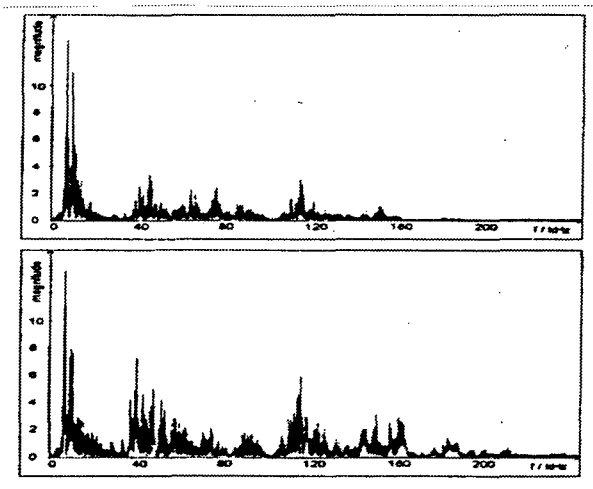

Figure 2.13 Spectrum of the vibration signal of the work tool holder (Top: sharp cutting edge. Bottom: worn out cutting edge [34].)

Figure 2.14 shows a five dimensional data set being mapped into to two dimensions, consisting of patterns belonging to 80 sharp and 80 worn out cutting edges under varying process parameters. The good separation between sharp and worn out cutting edges and the influence of the process parameters can be seen.

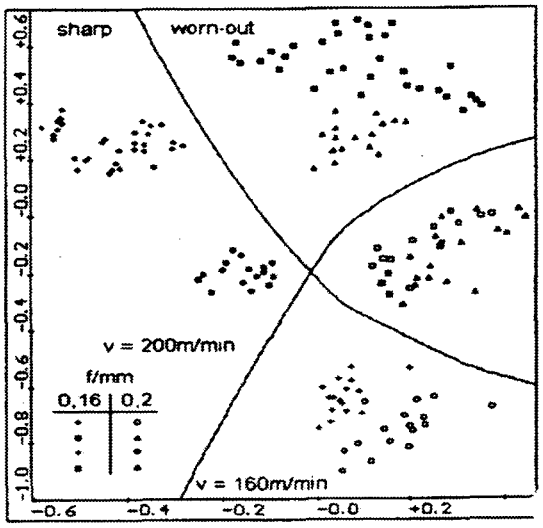

Figure 2.14 Sammon Plot for a data set of 80 sharp and 80 worn out cutting edges [34]

The neural network implementation was then introduced to use the obtained experimental data. Artificial neural networks can be used effectively for sub-symbolic feature processing, utilising its learning capabilities and generalisation of uncertain data. Several network models were analysed in order to ascertain their suitability for tool state classification and modeling of the wear process, and a Condensed Nearest Neighbor Network (CNNN) model was used. The CNNN model was developed by the Institute of Electrical Measurement, at the University of Paderbon. It is based on the condensed cluster principle. It is suitable to model complex cluster structures yielding a reclassification rate of 100%. The structure of the network consists of three layers, the size of the input and output layer corresponds with the number of features *J*, and the number of classes *K*, respectively. The topology of the network is shown in figure 4.

During the training phase the number of processing elements in the hidden layer is adjusted in order to fully map a given feature of clustering. Each hidden element

describes a hyperellisoid shaped subclass in the j-dimensional feature space. All hidden layer elements of one class are connected to the output element of the same class with a logical OR-function.

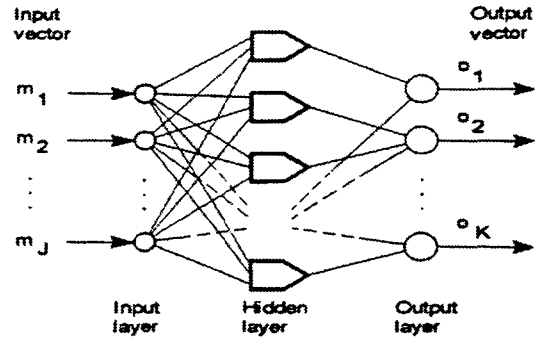

Figure 2.15 Topology of the CNN Network [34]

In the classification phase two cases have to be distinguished. If the presented pattern is located within the subclass area, only one output element is active and the pattern is classified to the class of this element. In case the pattern to be classified is positioned outside every subclass, the nearest cluster region is searched and the pattern is assigned to this class.

The calculated patterns of measured vibration signals of 5 tool life cycles were subdivided into four classes corresponding to the wear mark factor (WF) shown in table 2.1.

Class	**Wear Factor**
Class 1	WF<140µm
Class 2	170<WF<210µm
Class 3	230<WF<270µm
Class 4	WF>300µm

Table 2.1 Wear Factor classes

Only patterns that belonged to those classes, where used in the classification phase. The wear of the cutting edge is a continuous value and the transition between the four classes results in a considerable classification error. Therefore, several patterns

were classified to neighboring classes. The following table shows the class exchanges of the test by the CNNN-classification

	result of classification			
target	1	2	3	4
1	27	3		
2	5	20	3	
3		1	17	4
4				20

Table 2.2 Class exchanges [34]

By the use of a Multilayer Perception Network with back propagation learning rule (MLP) (using a least mean square error in conjunction with a gradient descent technique), the wear mark and crater depth can be estimated simultaneously without subdivision into several classes. With the same features, that were used in the four class problem, and a network consisting of five hidden and two output elements (figure 2.16) an estimation of wear mark with mean error of 13% referred to the maximal wear mark factor is possible. The estimation of crater depth for the same test set results in a mean error of 11% referred to as the maximal value.

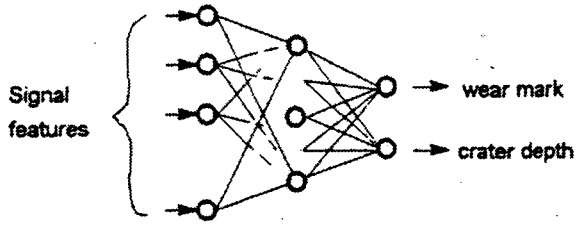

Figure 2.16 MLP structure used in the estimation of wear mark and crater depth [34]

The estimation results of wear mark and crater depth of a tool life cycle for a test set is shown in figure 2.17.

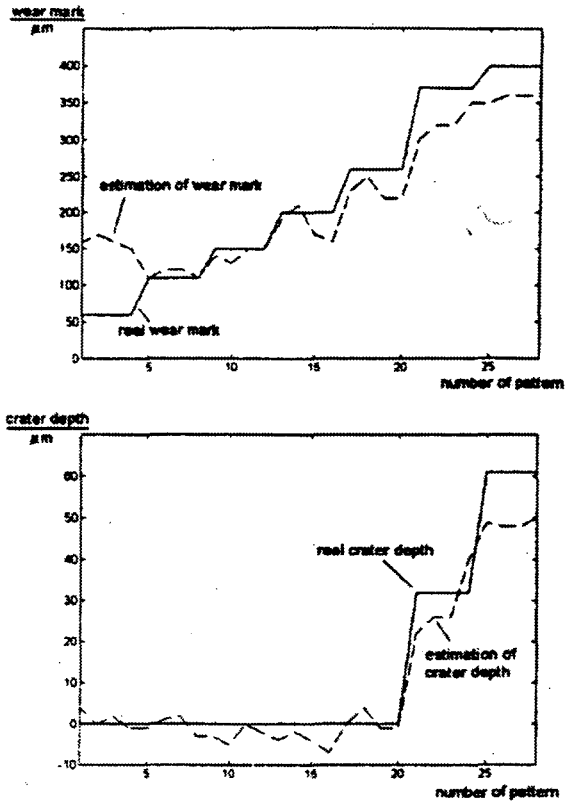

Figure 2.17 Estimation of wear mark (top) and crater depth (bottom) for the tool life cycle by an MLP [34]

The previous field of study is very relevant this current work. Data will be obtained through the direct testing methods, and then an indirect model formed with the aid of back propagation neural network technology. The main difference is that the previous field of study restricted its input variable, even to the stage where depth of cut was kept constant. Secondly the above study concentrated mainly on the wear of the cutting tool, and placed little emphasis on the effect that worn cutting tool has on the output surface quality.

B. Sick, and A. Sicheneder [45] have adopted a similar approach, in their research of time-delay neural networks for on-line tool wear classification and estimation in turning.

The increasing automation in metal-cutting manufacturing requires reliable and robust tool and surface monitoring systems. The supervision of a systems state for a just in time exchange is the most difficult task in this context. Based on an in-process

acquisition of signals with multisensor systems, it is possible to classify or estimate wear by means of artificial neural networks. The paper shows how a classification or estimation can be improved significantly considering the temporal development of wear by means of time-delay neural networks. The input parameters of the networks are process-specific parameters (like feed rate or cutting speed) and features extracted from signals of the multisensor system. This field of study is also very relevant to this current project. However once again the training for the neural network model are based on few machining variables and therefore the accuracy and variability of the model is questionable.

In a study [46,47] of neural network applications in on-line monitoring of turning by G. M. Zhang, and R. G. Khanchustambham, the performance variability of the intelligent process has been improved.

The need to improve quality and decrease scrap rate while increasing the production rate is motivating industry to consider untended machining as viable alternative. On-line monitoring of a machining process is the key component to success for an untended machining operation. The work is a framework for sensor based intelligent decision-making systems to perform on- line monitoring. Such a monitoring system interprets the detected signals from the sensors, extracts the relevant information, and decides on the appropriate control action. Emphasis is laid on applying neural networks to perform information processing, and to recognize the process abnormalities in a machining operation. A prototype monitoring system is implemented to demonstrate the working mechanism. For successful implementation of the developed intelligent monitor, an instrumented force transducer is designed for signal detection and is used in a real time turning operation. A neural network monitor based on feed forward back-propagation algorithm is developed and tested under the machining of advanced ceramic materials and steel. The monitor is trained by the detected cutting force signal, the measured surface finish, and the observed tool wear. The superior learning and noise suppression abilities of the developed monitor enable high success rates for monitoring in machining process.

For a successful implementation of untended machining, a better understanding of the machining processes and the functions they perform is required. This necessitates the development of sensors and intelligent decision-making systems.

In the study, a framework for sensor-based intelligent decision-making systems to perform on-line monitoring is proposed. Such a monitoring system interprets the detected signals from the sensors, extracts the relevant information, and decides on the appropriate control action. Emphasis is given to applying neural networks to perform information processing, and to recognize the process abnormalities in a machining operation. For signal detection, an instrumented force transducer is designed and implemented in real time turning operation. A neural network program based on feed forward back-propagation algorithm is developed. The program is tested by the simulation data, and verified by the experimental data.

It is evident that the development of hardware neural networks provides the monitoring system with fast computational capabilities. The advances in sensor technology enables inexpensive sensors to be easily mounted for monitoring the machining process. All these evidences justify that neural networks will become an attractive modeling tool for use in on- line monitoring of machining processes in the near future.

George Cooper [48] embarked on a research using the hybrid architecture of neural networks and fuzzy logic to help monitor and diagnose machining conditions. The strength of the neural network is the ability to filter out noise and their robustness in changing tool conditions that were not pre-programmed.

Extensive research needs to be conducted on combining sensors with an open architecture CNC machine tool controller. Simple sensors, such as tool breakage monitors, are already part of a machine tool control system. However, given the recent advances in sensor and vision technology the possibilities for developing new and sophisticated machine tool sensors are tremendous. The University of Leeds has performed some preliminary work combining video cameras with CNC lathes but more research is necessary, particularly using 3D vision and more sophisticated software.

Another attempt was made using image processing techniques to measure tool wear, using sophisticated approaches to capture and analyse the grey level images of tool tips. Parameters can then be used to predict wear; for example, surface texture and subsequently wear land morphology.

The above section details the amount of work that has been achieved via neural network prediction modeling. It is clear however, that a reliable system of predicting surface quality online, is yet to be developed. The current methods restrict their training data by limiting input variables and achieve inaccurate prediction models. Another area of research that looks at the surface quality, in a limited manner, is computer vision.

2.4.5 Optical Computer Vision

Optical computer vision involves similar ideas to that of offline tool wear identification, however it aims to provide an online determination of cutting conditions. Online determination through visual methods is still fairly approximate, as the vision is interfered with by the interaction of the work piece, the cutting fluids, and the chip formation. Therefore most online techniques of vision are generally used in conjunction with other measurement parameters, and intelligent systems such as neural networks and fuzzy logic.

Jong-Jin Park, and A. Galip Ulsoy [49,50] have studied on-line flank wear estimation using an adaptive observer and vision. The problem of developing a reliable on-line flank wear measurement system is treated using the integration of an adaptive observer, based on cutting force measurement, and computer vision. The flank wear is modeled as the summation of two unmeasurable states in a nonlinear dynamic system realized in state space equation form. The inputs to the system are the feed, the cutting speed, and the depth of cut (i.e., the cutting conditions) and the output is the cutting force. Based on a simplified version of this flank wear model, an adaptive observer is designed by combining the observer technique and the recursive least squares parameter estimation algorithm. The designed adaptive observer indirectly

measures the flank wear and simultaneously estimates one unknown model parameter, using measurements of the cutting force and the cutting conditions.

The adaptive observer is integrated with a computer vision system which can directly measure the flank wear with good accuracy. In the integrated system, the adaptive observer is intermittently calibrated using direct flank wear measurements via computer vision. The fundamental idea behind the proposed integrated method is that a less accurate indirect flank wear measuring method (i.e., the adaptive observer) is intermittently calibrated by a more accurate direct measurement method (i.e., computer vision).

A computer vision system is developed using an image processing algorithm, a commercially available computer vision system, and a microscopic lens. The developed algorithm is based on the difference between the intensity of the reflected light from a flank wear surface and that from the background. The difference is very significant and an appropriate selection of the intensity threshold level yields an acceptable binary image of the flank wear. This image is used by the vision computer for the calculation of the flank wear.

The flank wear model parameters that need to be known, are determined through several preliminary experiments, or from data available in the literature. Cutting conditions are selected to satisfy the assumptions made on the design of the adaptive observer. The resulting cutting conditions are typical of those used in finishing cutting operations. The integrated method is tested in turning experiments under both constant and time varying cutting conditions, and yields very accurate on-line estimation of the flank wear development. The flank wear model can then be incorporated to evaluate the corresponding surface finish quality.

This study generates a good comparability of the flank wear between experimental and theoretical results. The only problem is that for an expensive online method, it generates very little towards an on line implemented system for tool and surface wear. It is merely a study of the tool deterioration process, rather than a solution to the industry problem.

The real-time estimation of tool wear in machining operations is important for scheduling tool changing times and for adaptive process control and optimization. However, surface finish cannot be measured directly during cutting. An on-line estimation method has been developed by Galip Ulsoy [51-54], which uses force measurements during cutting. Unlike the above method where online vision is attempted while cutting takes place, direct optical measurement of tool wear using a computer vision system is used between cuts. This is almost an offline measurement system but given the other process variables being measured, it is essentially online analysis.

This approach permits the estimation of surface wear under varying cutting cuttings (e.g., changes in depth-of-cut, feed and speed), as would be required in many production situations including with adaptive process control. The method developed uses force measurement during cutting together with a process model. The model is the basis for an adaptive state estimator, which simultaneously estimates model parameters and the model states (i.e, state of wear of the tool and surface). This force based estimation is intermittently calibrated by the use of a computer vision system for direct optical wear measurement between parts. For production situations where a single tool edge produces many parts, the vision system can be used by itself. On the other hand, if a single part is produced by one or more tool edges then the force based system only can be used. The two are used in conjunction for production situations where a tool produces several.

The previous field of work exposes very little new ground. The computer vision used is essentially an offline technique and therefore the reliance is on the force measurements of the system for online integration. While this may be acceptable, it governs only a small section of the overall machining performance (based on machining variables), and would give on online implemented system such as a neural network, very little training. Hence inaccurate approximations may be expected from the final online intelligent system.

The optical and computer vision technique is considered an unreliable quantitative prediction model, because of its impracticability online. Metallurgy displays a

similar outcome due to its restrictive implementation. This is discussed in the next section.

2.4.6 Metallurgy

Metallurgy is the science and technology of metals. It studies the elasticity, plasticity, dislocation theory, and strengthening mechanisms associated with metals. This research area also investigates the testing and controlling of mechanical properties and their variations with temperature, strain rate, and microstructure.

Almost all engineering materials (except inorganic glasses and organic polymers) are polycrystalline. This means that they consist of any number of small crystals, which touch each other along grain boundaries. Tool materials consist of two or more phases, each having a different chemical composition and often a different crystal structure. The mechanical properties of tool metals and alloys are strongly influenced by such parameters as the size, shape and orientation of the grains in polycrystalline single-phase materials, and also by the size, shape, spacing and distribution of second phase particles in multi-phase materials [55-57]. It is necessary, therefore, to be able to characterize the microstructure of materials in a quantitative manner. This can be done using technique known as quantitative metallography (also called qualitative microscopy or stereology). Therefore by qualitative metallurgy the flank and crater wear of the tool can be determined. The process can also be used to determine the quality of the surface finish, but because of measuring conditions, can only be achieved offline. This process is usually in conjunction with other online monitoring techniques.

Metallurgy is an expensive and unproductive form of surface analysis. Given that only one variable result is achieved through experimentation, it gives very little useful data, and very little potential for the use with a smart system. Metallurgy is very similar in this manner to another restrictive measurement process, radiography.

2.4.7 Radiography

Radiography uses a broad range of x-ray energies to evaluate materials and components. Radiography covers a broad range of materials, components, and assemblies. Typical inspections include x-ray and gamma-ray energies from 3 keV to 9 MeV using both film and electronic imaging techniques. Current and future emphasis is on quantitative and micro evaluations based on a broad range of x-ray energies and beam geometry's coupled with sophisticated digital signal and imaging processing.

In the broadest sense, radiography is a nondestructive technique for evaluating the internal characteristics of a specimen using some form of penetrating radiation. The data is presented as a two-dimensional image in analog form, but the data may be processed and stored in the digital domain.

Radiographic applications fall into two categories [58], firstly the evaluation of material properties and evaluation of manufacturing. This includes the determination of composition, density, uniformity, and cell or particle size. Secondly are the assembly properties. Manufacturing and assembly property evaluation is normally concerned with dimensions, flaws (voids, inclusions, and cracks), bond integrity (welds, brazes, etc.), and verification of proper assembly of component pieces. Broad inspection capabilities.

Radiography is routinely used to inspect machined and cast components for voids, cracks, and inclusions. Low-density foam and composite materials are inspected for composition, density, uniformity, and foreign materials.

W. L. Kanizar, D. Liu, K. S. Moon, and J. W. Sutherland [59] have investigated radiographic techniques. The surface finish of a turned part is primarily generated from process parameters such as feed, tool geometry, depth of cut, and cutting speed. A micropositioner system utilizing a magnetostrictive material, Terfenol-D, as a linear motor is presented as a means to actively control the process. The system has an actuator clamped in a flexor that is rigid in the feed and main cutting force

directions, yet is flexible in the radial direction. Using control algorithms implemented on a digital computer, the system can provide a means to compensate for deleterious vibrations. The system has also been used to manipulate the tool position in the radial direction so that non-circular turning can be accomplished.

Radiography and similar procedures are still mainly used for surface quality checking offline. While reliable, by the time the component is checked the lost time, material, and hence capital is extremely high. Therefore the need for the technology employed in this thesis. This highlights the need for the need for the implementation of an online smart system. In the next section, another smart system, similar to, but not as widely utilised as neural networks, is discussed.

2.4.8 Fuzzy Logic

Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. The idea of fuzzy logic was first advanced by Lotfi Zadeh in the 1960s. Zadeh was working on the problem of computer understanding of natural language. Natural language (like most other activities in life and indeed the universe) is not easily translated into the absolute terms of 0 and 1. (Whether everything is ultimately describable in binary terms is a philosophical question worth pursuing, but in practice much data we might want to feed a computer is in some state in between and so, frequently, are the results of computing [60].)

Fuzzy logic includes 0 and 1 as extreme cases of truth (or "the state of matters" or "fact") but also includes the various states of truth in between so that, for example, the result of a comparison between two things could be not "tall" or "short" but ".38 of tallness." Fuzzy logic seems closer to the way our brains work. We aggregate data and form a number of partial truths, which we aggregate further into higher truths, which in turn, when certain thresholds are exceeded, cause certain further results such as motor reaction.

Tae Jo Ko, and Dong Woo Cho [61] have studied this principle through surface wear monitoring in diamond turning by fuzzy pattern recognition. While diamond tools

were used in this experimentation the procedure used would give similar output for stainless steel or carbide tools. The paper introduces a fuzzy pattern recognition technique for monitoring single crystal diamond tool wear in the ultra-precision machining process. Selected features by which to partition the cluster of patterns were obtained by time series AR modeling of dynamic cutting force signals. The wear on a diamond tool edge appears to be classifiable into two types, micro-chipping and gradual, both very small compared to conventional tool wear. In this regard, a fuzzy technique in pattern recognition was used, which considers the ambiguity in classification as well as the weakness of the cutting force variation to monitor the diamond tool wear status.

There are multitudes of signal sources to establish an in-process control for the hard turning process that are also largely applicated in conventional turning processes. Acoustic emission analysis, shear force and effective power measurement as well as the spindle speed control belong to these techniques. An evaluation strategy is currently being developed by a useful combination and integration of sensor signals in a signal processing program with integral Fuzzy-Logic toolboxes that allows a reliable and in time detection of defects within the process.

The applicability of signals for a process control in hard turning is being checked in systematic cutting experiments. Interrupted and continuous cutting conditions are still to be investigated. The needed investigations will be carried out on a high precision turning machine. The CNC-control of this machine also allows, among the recording of the effective power, the evaluation of the closed loop position control without the operation of additional sensor technology.

A successful elaboration of this project can therefore be a good basis for intensive utilization of this technology also regarding its use in manufacturing of large lot sizes, yet in current industry neural networks are used much more readily. And found to be more advantageous in their data acquisition and modeling techniques. The fuzzy logic system is also largely considered qualitative rather than quantitative, where it still relies heavily on empirical models. Therefore the reason for a well trained neural network system for online implementation of tool wear analysis, and surface finish profiling.

The surface finish of a machined component at present can only be measured offline. There is a great need for an online method of measurement. This is where a smart system such as a neural network or fuzzy logic is required. These smart systems require extensive training to be able to achieve a quantitative estimation. The most common form of training is through force modeling, discussed next.

2.4.9 On-line Force Modeling

Force modeling of the turning system is probably the most common and most heavily researched form of online investigation. Usually force analysis through a dynamometer or other device, is incorporated with other measured variables, and used in conjunction with a smart system.

Figure 2.18 Experimental Set-Up for Dynamic Force Monitoring [63]

Research studies carried out by KS Lee, and KHW Seah [62], on tool wear monitoring (Fig 2.18) have shown that there is a good correlation between the dynamic cutting force and flank wear. A characteristic peak can be observed when the dynamic force is presented in the frequency spectrum as shown in Fig 2.19. The study has also shown that the amplitude of the dynamic tangential force reveals a trend whereby it increases as the flank wear increases, and decreases on approaching tool failure. Tests were performed on various work piece materials such as AISI 01, AISI 4340, AISI 4140 and AISI 1148. Work pieces using P30 tungsten carbide tool inserts suggest that a threshold value of the percentage drop in the dynamic

tangential force from its maximum can be used to signal the on-set of tool failure (Fig.2.20).

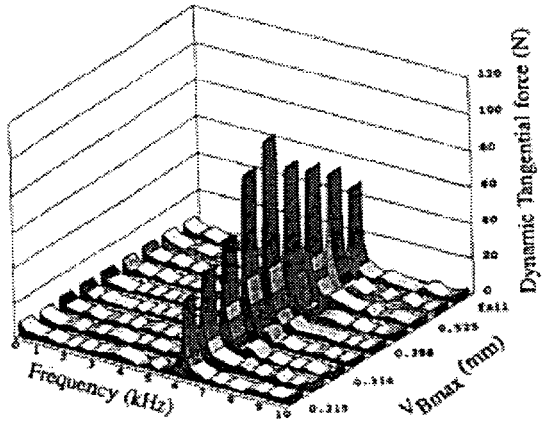

Figure 2.19 Frequency Spectra of the Dynamic Tangential Cutting Force [63]

Since the dynamic tangential force was observed to have a good correlation with the system wear, it is possible to use a personal computer to monitor the dynamic force during the machining process. A piece of on-line tool wear monitoring software has been developed to track the dynamic tangential force signals to give a pre-warning of imminent tool failure. A personal computer equipped with a Fast Fourier Transform card was used to perform the necessary data acquisition and FFT processing. The software developed so far allows for either continuous monitoring of tool wear or intermittent stoppages and monitoring. The results can be displayed in either the DOS or WINDOW environments.

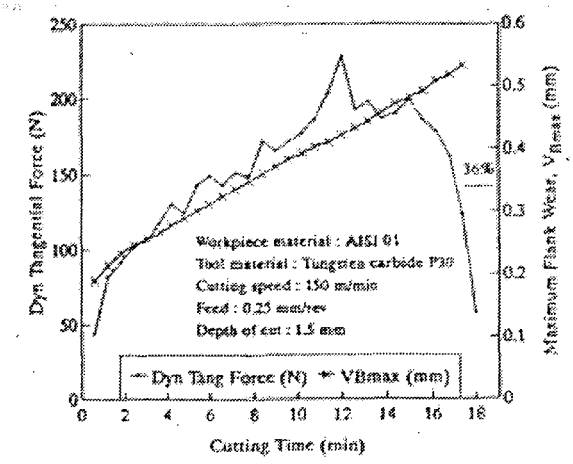

Figure 2.20 Illustration of the Dynamic Tangential Force and Max. Flank Wear for AISI 01 Steel [63]

Two criteria can be set to indicate the on-set of tool failure. The first criterion is reached when the percentage drop of the dynamic force from its maximum amplitude exceeds a predetermined threshold value. The second criterion confirms this when the gradient of the dynamic force curve exhibits a negative value. A pre-warning of tool failure will be activated and the machine may be stopped for tool change. This approach links the systems forces to the impending tool wear, and hence the region of surface quality can be predicted.

While an in depth force analysis has been carried out on the wear of the turning tool, this field of study has relied on one machining parameter only. Rather than using a smart tool, to alert of the condition of the surface finish, its still relies on a machine operator to monitor the output component quality.

2.5 CONCLUDING REMARKS

The previous section explains various attempts made by research groups to:

1. The study of tool conditions and surface finish.

2. Attempt to use a smart system to apply the trends into actual machining and to estimate on-line surface finish.

The previous methods investigated, work towards an approach for an on-line measurement system, but all lack essential characteristics. Most of the methods employed restrict themselves to too few process variables. With only a few machining parameters, the results of any testing can only be indecisive and inconclusive, especially over a broad range of cutting conditions. As test data can be essentially seen as training data for a smart system, previous methods of investigation have not provided the network model with a quantitatively reliable training set.

The previous investigations have not utilised the technology of smart systems to further increase the automation of the process. Currently it is not feasibly possible to

measure the quality of a surface finish online, and therefore a smart system is needed.

Given this problem, this field of study attempts to alleviate these problems by increasing the process variables, and given that appropriate training has been achieved, integrate a neural network system to "decision make" for the online system.

CHAPTER 3

NEURAL NETWORKS

In the previous chapter, the literature review, it was revealed that by using a smart system, such as a neural network, the performance prediction of the online monitoring system is maximised. In this project neural networks will be used to monitor the condition of a work materials surface finish, given any variable cutting condition. Firstly however it is important to clarify the complexity of how a neural network functions, and then how it will be incorporated into the project.

3.1 ARTIFICIAL NEURAL SYSTEMS

Artificial neurons are a fundamental part of artificial neural networks, forming the processing units for such systems. The artificial neuron is modelled on the basic concepts of the biological neuron. The first modelling of neurons dates back to the 1940s and was carried out by McCulloch and Pitts [103]. Drawing on their work on biological neurons, they proposed the following model [102]:

> "A synthetic neuron forms a weighted sum of the action potentials which arrive at it (each of these potentials is a numeric value which represents the state of the neuron which has emitted) and then activates itself depending on the value of this weighted sum. If this sum exceeds a certain threshold, the neuron is activated and transmits a response (in the form of an action potential) of which the value is the value of its activation. If the neuron is not activated it transmits nothing."

Similar to biological neurons, artificial neurons accept inputs from other similar neurons, process the inputs and send a single output to other neurons in the system. Likewise, artificial neurons communicate via weighted interconnects. The basic structure of an artificial neuron, shown in Figure 3.1, highlights the similarity

between artificial and biological neurons. The inputs to the neuron, shown as x_1 to x_i, for *i* inputs, represent dendritic connections in biological neurons, while the weighted connections, denoted as w_{j1} to w_{ji} for neuron *j*, represent synapses. In addition, every artificial neuron has one output line, representing the axon of a biological neuron, which can branch out to form connections with other neurons.

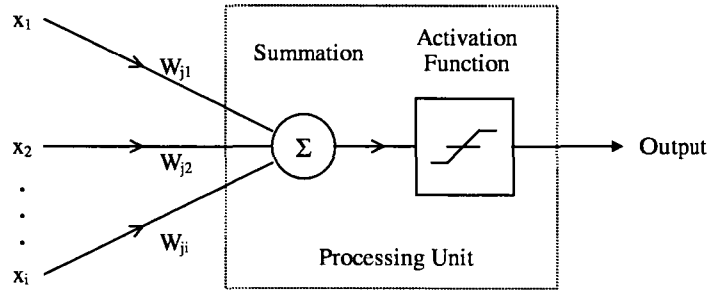

Fig. 3.1 Basic structure of an artificial neuron

The cell body of a biological neuron is represented as a processing unit in the artificial neuron model. There are essentially three functions that combine to give the neuron its processing capability; the input function, activation function, and output function. The input, activation and output functions of a neuron can usually be combined into one function, called the transfer function [64]. To understand the transfer function it is useful to consider these three functions separately. The input function performs a summation of the multiplication of inputs with the corresponding weights. It is an additive function that essentially obtains the summed input for each neuron by multiplying the output of every neuron connected to it by the associated synaptic weight for each connection and then summing the result. The net input to the *jth* unit, net_j, can be written as [68]:

$$net_j = \sum_i x_i w_{ji} \tag{3.1}$$

where, index *i* denotes the number of input neurons

The output from the input function forms the input for the activation function. The activation function is a nonlinear function that, when applied to the net input of a

neuron, determines the output of that neuron [69]. The activation function can be any function that is monotonically increasing and differentiable [73]. The values that the activation function can output is generally limited to either the range 0 to 1 or -1 to 1, depending on the activation function used. Early neural models used a simple threshold function, or step function, as the activation function. This particular type of activation function allowed a 1 to be output from the neuron if the weighted sum of the inputs exceeded the threshold, otherwise the output is 0. The threshold function is shown graphically in Figure 3.2(a).

More recently the threshold function has been replaced by a more general nonlinear sigmoid function, also called the logistic function [66]. A sigmoid function may be loosely defined as a continuous, real-valued function whose derivative is always positive, and whose range is bounded [69]. The logistic function is written as [106]:

$$f(net) = \frac{1}{1 + \exp(-net)} \tag{3.2}$$

A graphical representation of the logistic function, shown in Figure 3.2(b), highlights the bounded range of the function, 0 to 1.

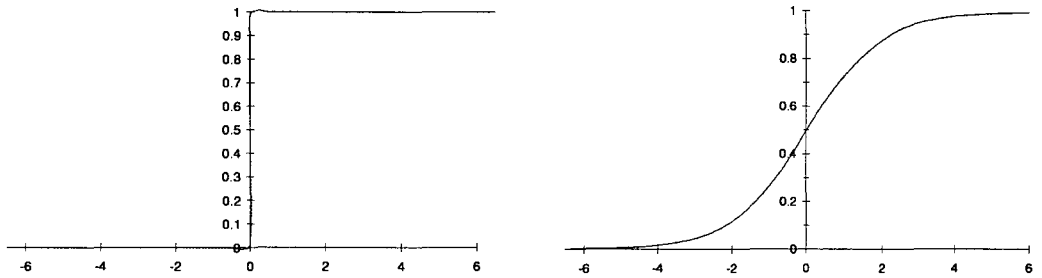

Fig. 3.2 (a) Threshold function, and (b) Logistic function

An advantage of this particular type of function is that its derivative, which will later be shown to be a significant aspect of neural computation, is easily calculated. In fact the derivative of the logistic function, f'(net), is written as [107]:

$$f'(net) = f(net) * (1 - f(net)) \tag{3.3}$$

The third component of the transfer function, the output function, is usually chosen to be equal to the output of the activation function, ie. the output of the neuron will be the same as the activation [64]. Hence, the logistic function shown in Figure 3.2(b) then represents the neuron transfer function with the horizontal axis representing the weighted summed input and the vertical axis representing the neuron output.

One of the simplest forms of a neural network is the perceptron, developed by Frank Rosenblatt in the early 1960s. The perceptron is a pattern classification system that recognises abstract and geometric patterns from optical input patterns, despite noise in the input [104]. The computation that takes place in the processing neurons of the perceptron is based on a simple principle. The neuron computes the weighted sum of the input signals and compares that net weighted input to a threshold value, T [65]. If the net input is greater than or equal to the threshold, the neuron outputs +1, if not, it outputs -1 [65]. Hence, the transfer function used in the hidden and output layer neurons of the perceptron is written as [65]:

$$I = \sum_{i=1}^{n} w_i x_i \tag{3.4}$$

$$y = \begin{pmatrix} +1, \text{if } I \geq T \\ -1, \text{ if } I < T \end{pmatrix}$$

where, I = net weighted input to the neuron,

w_i = component of weight vector,

x_i = component of input vector, and

y = output of the neuron

Rosenblatt introduced a training algorithm for the perceptron that provided the first procedure that could be used to allow a network to learn a task [65]. This training algorithm was used for changing weights in the network using the formula [65]:

$$w_{new} = w_{old} + \beta\, y\, x \tag{3.5}$$

$$\text{where, } \beta = \begin{pmatrix} +1, \text{ if the perceptrons answer is correct} \\ -1, \text{ if the perceptrons answer is wrong} \end{pmatrix}$$

y = the perceptron's answer,

w = weight vector, and

x = input vector

Frank Rosenblatt [64] and other researchers were able to demonstrate mathematically that the perceptron training algorithm can always solve any linearly separable problem in a finite number of steps. The perceptron learning rule convergence theorem states that if weights exist to allow the network to respond correctly to all training patterns, then the rule's procedure for adjusting the weights will find values such that the network responds correctly to all training patterns, ie. the network solves the pattern or learns the classification [109]. However, the perceptron never became a viable application system due to certain limitations associated with its ability. Nevertheless, the development of the perceptron laid some important groundwork and inspiration for other researchers to further develop neural networks to the stage they are today and therefore is seen as a historical landmark in the field of neural systems.

If we conceive each neuron in an artificial neural network as a primitive function capable of transforming its input to a precisely defined output, then artificial neural networks are nothing more but networks of primitive functions [104]. Artificial neural networks are used in many cases as a 'black box'. While it is desired that a certain input produce a particular output, how the network achieves the particular output is left to a self-organising process. In general, a neural network is used to map an *n*-dimensional real input (x_1, x_2, ..., x_n) to an *m*-dimensional real output (y_1, y_2, ..., y_m).

Neural networks are trained for particular applications such that they learn to solve associated problems, they are not programmed to do so. Hence, 'training' is fundamental to all neural networks and is the process of modifying the network interconnection weights. Successful training results in a real number being assigned to every interconnection in the network such that every interconnection has a unique associated weight and the performance accuracy of the network is maximum. By

varying the weights associated with each input, the neural network can, in conjunction with the transfer function, implement any transformation between its inputs and outputs [70]. However, these weights need to be computed for each particular application and, because it is not usually possible to compute them directly, this is achieved by the repetitive, and often time-consuming, process called 'neural network training' [70]. Algorithms for varying these connection strengths or weights such that learning ensues are called 'learning rules' [72]. The specific algorithms used for learning are dependent largely on the particular network model being considered, hence, the respective algorithms are discussed in relation to the network that they apply in later sections. Learning methods for neural networks may be broadly grouped as supervised and unsupervised, with a great many paradigms implementing each method [74]. Prior to discussing these two particular training techniques, however, it is useful to consider a taxonomy of neural networks, Figure 1.2.4. Neural networks, in addition to being classified by their training technique, either supervised or unsupervised, are classified as either feedforward or recurrent networks, as the respective categories reflect the processing behaviour of the network. Hence, a discussion of the difference between feedforward and recurrent network models is required followed by a discussion of the training techniques.

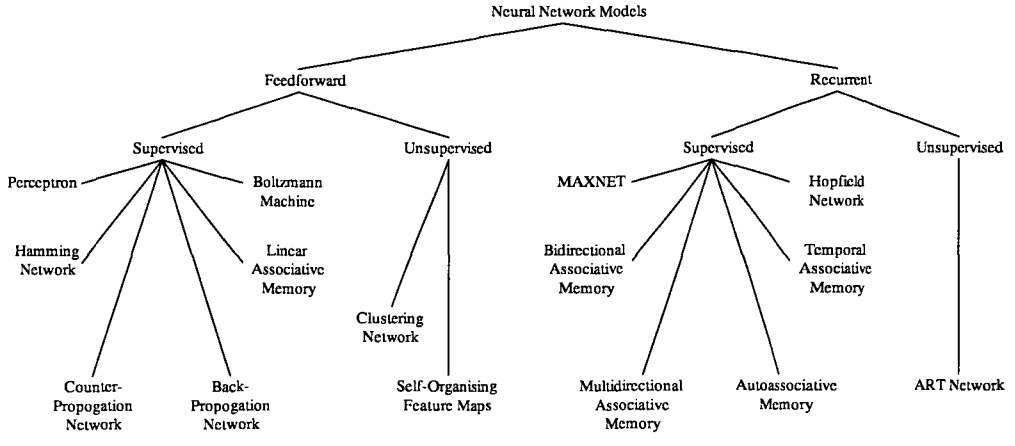

Fig. 3.3 A taxonomy of neural network models [75]

3.1.1 Feedforward Networks

In a feedforward network, input activity signals propagate in one direction only, from the input stage through intermediate neurons to an output stage. The basic feedforward network contains three distinct layer types; input, hidden and output. This type of neural network has one input layer, one output layer, and any number of hidden layers in between [76,77]. The basic architecture of a feedforward network is shown in Figure 3.4 with the main features labelled.

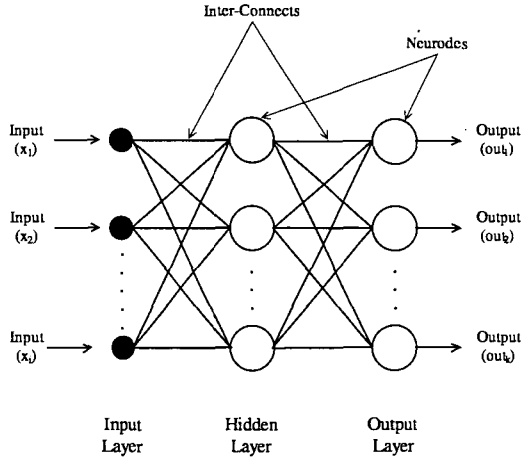

Fig. 3.4 Basic structure of a multi-layer feedforward network [66]

Each layer in the network contains neurons which receive any number of inputs, process those inputs and send a single output to other neurons in the subsequent layer. The neurons in the input layer receive information from a knowledge base while the output layer neurons send information to the surrounding environment. The neurons in the hidden layer are the processing units of the network.

3.1.2 Recurrent Networks

A recurrent neural network distinguishes itself from a feedforward neural network in that it has at least one feedback loop [115]. For example, a recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons, as illustrated in the architectural graph of Figure

3.5. Hence, due to the existence of feedback connections among neurons, a recurrent neural network exhibits dynamic behaviour. That is, given the initial state, the state of a recurrent neural network evolves as time elapses. If the recurrent neural network is stable, a state of equilibrium can eventually be reached [135]. Recurrent neural networks are usually used for storing information as associative memories and solving computationally intensive problems. A classic example of a recurrent network that learns through supervised training is the Hopfield network, introduced by John Hopfield [87,88].

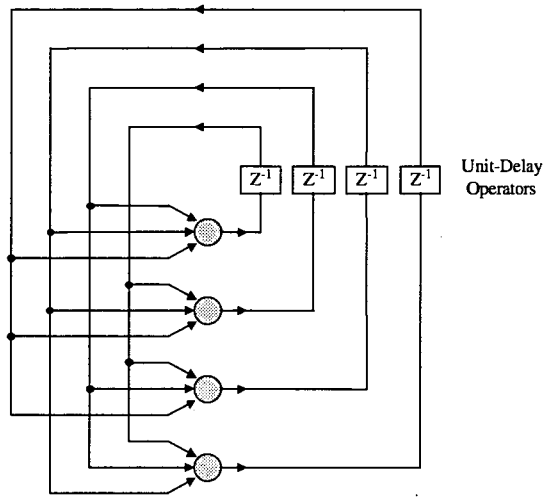

Fig. 3.5 Recurrent network with feedback loop [115]

3.1.3 Supervised Training

During supervised training a neural network adjusts its weight vector in the direction of minimising the error between its outputs and the targets to be learned [71]. Supervised training involves a teacher providing input patterns to the network that are expected to be encountered during operation and an associated output pattern that the network is expected to produce when it receives the particular input pattern. An iterative process is used in conjunction with this style of learning such that the network adjusts its weights continually until the error between the network output prediction and the target output is minimum. Once the minimum error is achieved the network weights remain unchanged and the process has converged. The process of updating the network weights once is referred to as an 'epoch' and the number of input-output patterns used for the training set is referred to as the 'epoch size'. Some

researchers use an epoch size of one, meaning that the weights are updated after each training case is presented, however, it is usually preferred to use the entire training set for each epoch as this favours stability in convergence to the optimal weights [69]. A distinct form of supervised training is reinforcement learning where the only feedback is whether each output is correct or not [78]. Reinforcement learning is a form of supervised learning because the network receives some feedback from its environment. However, the feedback is only evaluative, not instructive. The reinforcement signal is a single yes or no, it gives no hint as to what the right answer should be. Networks that can be trained using reinforcement learning can be either feedforward or recurrent.

3.1.4 Unsupervised Training

Unsupervised training differs from supervised training in that it only requires input vectors to train the network. During unsupervised training, network weights are adjusted so that similar inputs produce similar outputs [74]. For unsupervised training, the only available information is in the correlations of input data or signals. The network is expected to create categories from these correlations, and to produce output signals corresponding to the input category [66]. The system searches for similar features in the training inputs to group them into categories where the numbers of a single category share common features [108]. Unsupervised learning algorithms use patterns that are typically redundant raw data having no labels regarding their class membership, or associations, and in this mode of learning, the network must discover for itself any possibly existing patterns, regularities or separating properties [67]. In discovering these the network undergoes changes in its parameters, hence the name self-organisation often associated with these particular network models [89].

In addition to being either feedforward or recurrent, and the technique adopted to train the network, there are many properties and characteristics that distinguish one neural network from another. Moreover, an artificial neural network is generally defined by [114]:

- the network properties - network topology

	- types of connections
	- order of connections
	- weight range
• the node properties	- activation range
	- transfer functions
• the system dynamics	- weight initialisation scheme
	- activation-calculating formula
	- learning rule

It is convenient to visualise neurons as arranged in layers, where typically, neurons in the same layer behave in the same manner [109]. That is, the factors that determine the behaviour of the neuron, namely its activation function and the pattern of weighted connections over which it sends and receives signals, are the same within each layer. The arrangement of neurons into layers and the connection patterns within and between layers defines the network architecture, which is commonly either single-layer or multi-layer. Typically, the input layer of neurons is not included in the determination of the network layer size as the input layer performs no computation. It simply passes forward the input pattern presented to the network to neurons in the subsequent layer. Hence, a single-layer network consists only of an input and output layer while a multi-layer network has in addition any number of hidden layers between the input and output layers. It should be noted that as the input layer is not included when describing the network layer size, a network with N layers is denoted as an N-1 layer network.

3.1.5 Important Characteristics of Neural Networks

Some of the major characteristics of neural networks that make them potentially so useful are listed below [64].

- A neural network is composed of a number of very simple processing elements, or 'neurons', that communicate through a rich set of interconnections with variable weights or strengths.

- Memories are stored or represented in a neural network in the pattern of variable interconnection weights among the neurons. Information is processed by a spreading, constantly changing pattern of activity distributed across many neurons.
- A neural network is taught or trained rather than programmed. It is even possible to construct systems capable of independent or autonomous learning; some neural networks are capable of learning by trial and error.
- Instead of having a separate memory and controller, plus a stored external program that dictates the operation of a system as in a digital computer, the operation of a neural network is implicitly controlled by three properties: the transfer function of the neurons, the details of the structure of the connections among the neurons, and the learning law the system follows.
- A neural network naturally acts as an associative memory. That is, it inherently associates items it is taught, physically grouping similar items together in its structure. A neural network operated as a memory is content addressable; it can retrieve stored information from incomplete, noisy, or partially incorrect input cues.
- A neural network is able to generalise; it can learn the characteristics of a general category of objects based on a series of specific examples from that category.
- A neural network is highly fault tolerant. A neural network keeps working even after a significant fraction of its neurons and interconnections have become defective. Its performance degrades slowly and smoothly as neurons and interconnections fail.
- A neural network innately acts as a processor for time-dependent spatial patterns, or spatiotemporal patterns.
- A neural network can be self-organising. Some neural networks can be made to generalise from data patterns used in training without being provided with specific instructions on exactly what to learn.

The listing of characteristics given here is useful to summarise the basic functions of neural networks and highlight some of the desirable features that are making neural networks popular for a growing range of applications. Each of the characteristics listed play an important role in the overall performance of neural networks and will be considered in detail as they apply in following sections.

3.2 SOME CONSIDERATIONS FOR IMPROVING AND EVALUATING NETWORK PERFORMANCE

There are many important considerations when designing the neural network training data and architecture. In order to achieve optimum performance from the network there are some decisions to be made in regard to the data that is used to train the network and the preparation of that data as well as the particular network architecture to use. In addition, testing of the network after successful training is necessary to evaluate and assess the network performance.

3.2.1 Design of Network Training Data

In designing the training data for neural networks there are a number of important considerations for developing a satisfactory training set. The performance of a network depends heavily upon the vectors used to train it [74]. One of the most significant considerations is selecting the size of the training set. If the neural network is going to be effective at its ultimate task, the training set must be complete enough to satisfy two goals [69]:

i) Every variable in the training data set must be adequately represented. Usually, the training data will consist of several possible subgroups, each having its own central tendency toward a particular pattern. All of these patterns must be represented sufficiently.

ii). Within each class, statistical variation must be adequately represented. It is the presence of random noise imposed onto pure patterns that makes most neural network applications necessary. The training set must be designed to insure that an adequate variety of noise effects are included.

A relationship among the number of training patterns, P, required, the number of weights, W, to be trained and the accuracy of the classification expected, e, is given by the following relationship [94]:

$$\frac{W}{P} = e, \quad \text{or rewritten,} \quad P = \frac{W}{e} \tag{3.6}$$

If the network is trained to classify the fraction 1 - (e/2) of the training patterns correctly, where $0 < e \leq 1/8$, then in order to be sure the network will classify 1 - e of the testing patterns, there must be sufficient training patterns such that the network is able to generalise as desired. In selecting sufficient training patterns, Equation 1.17 is useful. Commonly, e is chosen to be about 0.1, meaning the number training patterns required is approximately 10 times the number of network weights, giving confidence that the network will classify 90% of the testing patterns correctly, assuming it was trained to classify 95% of the training patterns correctly.

3.2.2 Normalising Network Input

Neural network performance can often be improved if the data set is modified by removing insignificant characteristics. The important aspects of the training data are often independent of the value of offsets and standard deviations, these may merely obscure the issue and complicate the network's task [23]. Hence, scaling the network input, or normalising, is one such method of removing insignificant characteristics from the training set by scaling the magnitude of each input vector component between some predetermined limits. Uniform scaling of the input data results in the individual components of the input vector being recognised by the network to be of equal magnitude. Consequently, the network is not bias towards components in the input vector that are of a higher magnitude. Furthermore, normalisation of target output data used for training the network is recommended as most training algorithms minimise the total error of all outputs. If target output variables are unequally scaled, those with larger variabilities will be favoured as they will dominate the error sum [69]. Moreover, normalisation of network input data will increase the numerical stability of the network data process [83].

Normalisation of target output data is mandatory when the output of the network is bounded due to the particular activation function used. One specific example of this is the sigmoidal function whose output has been shown to be limited to the range 0 to 1. For this particular function the output target values that the network is learning

from must also be scaled to the bounded range to allow the network to learn the data. A common method of scaling, or normalising, data between the range 0 to 1 for a particular set of values of any magnitude is by the formula:

$$\text{norm}(x_i) = \frac{x_i - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{3.7}$$

where, $\text{norm}(x_i)$ = normalised *ith* value in a set of j values,

x_i = original *ith* value in a set of j values,

$\min(x_j)$ = original minimum value in a set of j values, and

$\max(x_j)$ = original maximum value in a set of j values

Through use of this method the data is scaled over the range 0 to 1 such that when x_i equals $\min(x_j)$, $\text{norm}(x_i)$ equals 0, and when x_i equals $\max(x_j)$, $\text{norm}(x_i)$ equals 1. Consequently, the remainder of x_i values between $\min(x_j)$ and $\max(x_j)$ are scaled uniformly between the bounded range 0 to 1.

3.2.3 Network Testing and Performance

While the training set is used to train the network, the test set is used to assess the performance of the network after training is complete [85]. For neural learning to be successful, it is essential for the system to perform correct classification of test samples on which the system has not been trained [111]. When training is complete it is interesting to try patterns not in the training set to see whether the network can successfully generalise what it has learned [66]. Generalisation in neural networks may be viewed as multi-dimensional interpolation [74]. To understand this, it is useful to consider a one dimensional input vector. For a given set of samples in the training set the individual values form some underlying, or unknown, curve. After training is complete the presentation of an input vector not included in the training set will require the network to generalise. In neural network generalisation the network will interpolate between points at the extremes of the input vector group so that given an intermediate value, x, the network can determine output values, $y = f(x)$, that lie on the unknown curve [113].

3.2.4 Selecting Network Architecture for Optimum Performance

One of the most important attributes of layered neural network design is choosing the network architecture [67]. The network architecture is a very important consideration for the optimal trainability and generalisation ability [112]. For feedforward network models this decision involves the selection of how many hidden layers are necessary and how many neurons are required within each hidden layer. Generally, the number of neurons in the input layer is equal to the dimension of the input vector to be classified, generalised or associated with a certain output quantity. Similarly, the number of neurons in the output layer is equal to the number of required outputs. For example, the number of required outputs could be the number of possible classifications for a given set of inputs or the number of parameters to be predicted by a network.

However, the decision of how many hidden layers and hidden layer neurons is more complex. For feedforward networks, it has been proven that there is no theoretical reason to ever use more than two hidden layers [69] and for the vast majority of practical problems it is rarely necessary to use more than one hidden layer [74]. The use of two hidden layers is usually only necessary in practice when the network must learn a function that has discontinuities [69]. The decision of how many hidden layers to use is quite critical as the use of a second unnecessary hidden layer can dramatically slow network training. This is due to two effects [69]:

1. The additional layer through which errors must be backpropagated makes the gradient more unstable. The success of any gradient-directed optimisation algorithm is dependent on the degree to which the gradient remains unchanged as the parameters change.
2. The number of false minima usually increase dramatically, meaning that there is a higher probability that after many time-consuming iterations, the network will be stuck in a local minima, resulting in the need to restart training.

It is strongly recommended that one hidden layer be the first choice for any practical feedforward network design and if using a large number of hidden layer neurons does

not satisfactorily solve the problem, then it may be worth trying a second hidden layer and possibly reducing the total number of hidden layer neurons [69].

Choosing an appropriate number of hidden layer neurons is extremely important as using too few will starve the network of the resources it needs to solve the problem, while using too many will increase the training time and may cause a problem known as overfitting [18]. In Figure 3.6(a) a good fit to the training data set is demonstrated, while in Figure 3.6(b) overfitting has occurred, as may be the case if too many hidden layer neurons are used. The circle in the figure represents a test set that the network may encounter. For a good fit, reflecting only a minimum number of hidden layer neurons interpolation is reasonable, however, for overfitting, reflecting the use of too many hidden layer neurons, interpolation is poor.

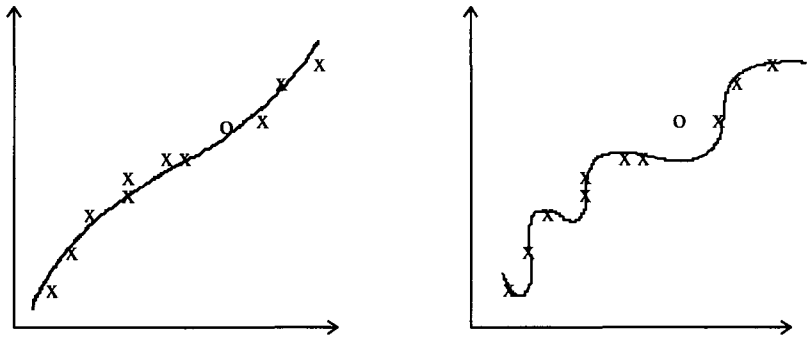

Fig. 3.6 (a) A good fit to noisy data, and (b) Overfitting of the same data [66]

A guideline for selecting the optimum number of hidden layer neurons is to use as few as possible to obtain a satisfactory solution. A common method for determining the minimum number of hidden layer neurons required is to compare the root-mean-square error of the network for a range of neurons. The minimum number of neurons that can be used without increasing the associated network error represents the number of neurons that should be used.

3.3 PARTICULAR MODELS OF THE NEURAL NETWORK PARADIGM

In this section the specific details of the particular network models that were studied and applied as a part of this research project are given. For each of the models the associated architecture, algorithms and particularities of the network type are provided. The particular models selected in this instance are:

- Widrow-Hoff network
- backpropagation network
- radial basis function network
- Kohonen network, and
- general regression neural network

The particular models listed are feedforward type networks that learn using a supervised training technique, with the exception of the Kohonen network, which uses the unsupervised training technique. While the majority of manufacturing applications use the backpropagation network, due to its proven success in many areas of manufacturing, alternative networks such as the Widrow-Hoff, radial basis function and general regression neural network were selected in order to provide a comparison with the more popular backpropagation model. In addition, the Kohonen network is used in conjunction with the radial basis function network to cluster the inputs for the radial basis function network, in order to minimise the number of hidden layer nodes required. The operating mode and characteristics of the Kohonen network make it suitable for this particular task and hence, selection as a model for this particular research project. Furthermore, considering typical applications in the aluminium industry involving process control, prediction and optimisation, the particular models of the neural network paradigm listed here are most appropriate for modelling the majority of applications. The Widrow-Hoff, backpropagation and radial basis function networks perform well, and have proven documented success, with continuous mapping applications and were chosen because of this. Furthermore, the general regression neural network was selected due to its modelling capability in addition to the fast training times available with this particular model. Nevertheless,

if the selected neural network models are found to be inadequate for the applications encountered in the aluminium industry then further models will be developed as required.

3.4 WIDROW-HOFF NETWORK, OR ADALINE

The adaline, or *ada*ptive *li*near *ne*uron, was introduced by Widrow and Hoff [87] in the early 1960s and is an adaptive patten-classification machine. The adaline learns through supervised training, using a learning law called the 'delta rule'. This learning rule minimises the mean squared error between the activation and the target value. During training, the adjustable weights connecting the input units to the output unit are updated to minimise the error between the predicted output and the target output supplied to the network. The delta rule for adjusting the *i*th weight, w_i, for each activation pattern, is written as [109]:

$$\Delta w_i = \alpha(t - y\_in)x_i \tag{3.8}$$

where, α = learning rate,

t = target output,

y_in = net input to output unit *y*, and is given by:

$$y\_in = \sum_{i=1}^{n} x_i w_i \text{, and} \tag{3.9}$$

x = vector of activations of input units

Hence, the objective of the delta rule is to find an optimum set of synaptic weights, w_1, w_2, ..., w_i, that minimise the error between the desired target and the corresponding network output.

3.4.1 Architecture

In its simplest form, this network consists of a single neuron along with its associated input interconnects and synapses [64]. The architecture of the adaline is shown in

Figure 3.7. An adaline is a single unit that is capable of receiving input from several units. A bias unit is also necessary, hence, an input with a constant value of +1 is supplied to the network with each activation, connected to the output unit through a weighted connection, *b*. Hence, the basic adaline has one output unit and *i* input units.

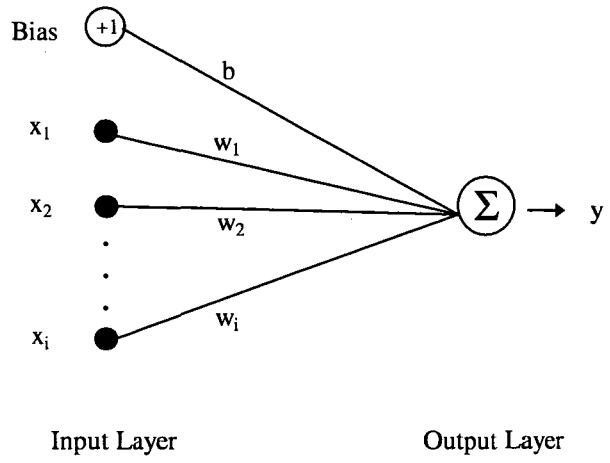

Fig. 3.7 Basic architecture of an adaline

Several adalines that receive signals from the same input units can be combined in a single layer network. However, if any number of adalines are combined so that the output from some of them becomes input for others of them, then the network becomes multi-layer, known as a 'madaline', short for many adalines.

3.4.2 Algorithm

A training algorithm for the adaline is as follows [109]:

Step 1. Initialise network weights, w_i, to small random values.
Set learning rate, α.
Step 2. While stopping condition is false, do Step 3.
Step 3. For each bipolar training pair, *s:t*, do Steps 4 to 7.
Step 4. Set activations of input units, $i = 1, \ldots, n$.

$$x_i = s_i \tag{3.10}$$

Step 5. Compute net input to output unit *y*:

$$y\_in = b + \sum_i x_i w_i \tag{3.11}$$

Step 6. Update bias, *b*, and weights, $i = 1, \ldots, n$:

$$b(new) = b(old) + \alpha(t - y\_in) \tag{3.12}$$

$$w_i(new) = w_i(old) + \alpha(t - y\_in)x_i \tag{3.13}$$

Step 7. Test for stopping condition:

If the largest weight change that occurred in Step 3 is smaller than a specified tolerance, then stop, otherwise continue.

In order to determine the success and sufficiency of supervised training using the delta rule it is necessary to have a quantitative measure of learning. As this supervised training algorithm involves the reduction of an error value then it follows that an error value be used to evaluate network training. Hence, the root-mean-squared, RMS, error is an adequate and commonly used error measure. RMS error is computed using the formula [85]:

$$\text{RMS Error} = \sqrt{\frac{\sum_p \sum_j (t_{jp} - a_{jp})^2}{n_p n_k}} \tag{3.14}$$

where, t_{jp} = target output for output neuron *j* after presentation of pattern *p*,

a_{jp} = the output value produced by output neuron *j* after presentation of pattern *p*,

n_p = number of patterns in the training set, and

n_k = number of neurons in the output layer

The RMS error is a useful measure of how close a network is getting its predictions to its target output values. For successful training the RMS error will decrease significantly in the initial stages of training and converge after a sufficient number of iterations have been completed. Generally, an RMS error value less than 0.1 indicates that a network has sufficiently learned its training set [85].

3.4.3 Applications

Although the adaline is a relatively simple network it has proven success in a variety of applications, including finance and investment [142] and weather forecasting [135]. However, the adaline has one major difficulty that limits its use, it can only be applied successfully to linearly separable pattern classification problems [64].

3.5 BACKPROPAGATION NETWORK

Backpropagation is the most widely used of the neural network paradigms and has been applied successfully in applications studies in a broad range of areas [78, 99, 100, 101]. Backpropagation networks are multi-layered feedforward neural networks that are trained using the error backpropagation procedure, a supervised mode of training. Backpropagation is a systematic method for training multi-layered artificial neural networks and is a form of supervised training [71].

3.5.1 Architecture

The architecture of a backpropagation network, shown in Figure 3.8, consists of an input layer, one or more hidden layers and an output layer. There are i input nodes, j hidden nodes and k output nodes. All input nodes are connected to all hidden nodes through weighted connections, w_{ji}, and all hidden nodes are connected to all output nodes through weighted connections, w_{kj}.

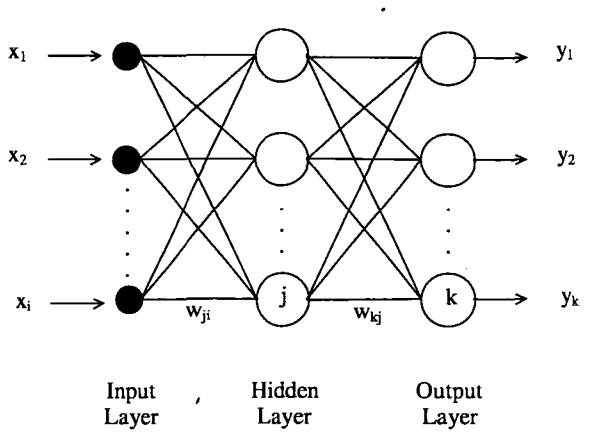

Fig. 3.8 Basic structure of a backpropagation network [66]

The sole function of the input neurons is to pass forward input patterns to neurons in the hidden layer. In this type of feedforward network there are no connections leading from a unit to units in previous layers, nor to other units in the same layer, nor to units more than one layer ahead [66]. Hence, every neuron in each layer communicates only with neurons in the immediate following layer. Only the hidden layer and output layer neurons complete any type of processing. The neurons in these layers perform three functions; an input function, an activation function and an output function. The input function performs a summing of the inputs and synaptic weights. It is a linear function and is given by the equation:

$$\text{net}_j = \sum_j x_i w_{ji} \tag{3.15}$$

where, net_j = weighted summed input to neuron j,

x_i = input i to neuron j, and

w_{ji} = weight connecting input neuron i to hidden layer neuron j,

The most common activation function used in backpropagation networks is the logistic function, which is a nonlinear sigmoidal function given by the following equation:

$$f(\text{net}) = \frac{1}{1 + \exp(-net)} \tag{3.16}$$

Hence, the input function summation is used in the processing performed by activation function. The purpose of the output function is to pass forward the output of the activation function. Hence, the output function is a linear function, it is equal to the output of the activation function.

Each training pattern presented to a backpropagation network is processed in two stages. In the first stage, the input pattern presented to the network generates a forward flow of activation from the input to the output layer. In the second stage, errors in the network's output generate a flow of information from the output layer backward to the input layer [64]. The error backpropagation procedure uses a gradient descent method which adjusts the weight in its original and simplest form by an amount proportional to the partial derivative of the error function with respect to the given weight [75]. The calculation of associated error for a given input pattern is determined only after the forward propagation of the pattern is complete. That is, for each neuron in the output layer of the backpropagation network a single real number is output, which is compared to a target value presented with each input pattern. The error associated with this comparison is then used to update the weights for all interconnections from the hidden layer to the output layer. Similarly, an error value is calculated for all neurons in the hidden layer immediately prior to the output layer, and subsequently, all weights are updated for interconnections that form inputs to this hidden layer. This process is completed until the last layer of weights has been updated in this manner. The error value, δ, is simple to compute for the output layer and somewhat more complicated for the hidden layers [85]. Considering firstly the output layer, for neuron *j* in the output layer, the error value is [85]:

$$\delta_j = (t_j - a_j)f'(net_j) \tag{3.17}$$

where, t_j = target value for unit *j*

a_j = output value for unit *j*

$f'(x)$ = derivative of the sigmoid function, and

net_j = weighted sum of inputs to *j*

Hence, the quantity (t_j - a_j) represents the difference between the target output and the network prediction while the derivative of the sigmoid function is used to scale the error. For the commonly used sigmoid function, the maximum value of the derivative corresponds to the point of maximum slope on the function curve. It is useful to reproduce a graph of the logistic function here along with a graph of the derivative of this function in order to highlight this important relationship, Figure 3.9. It can be seen that by including the derivative term in the error value calculation the error is scaled to make a larger correction when the weighted sum of the inputs is small, close to zero, and a smaller correction when the weighted sum of the inputs is large.

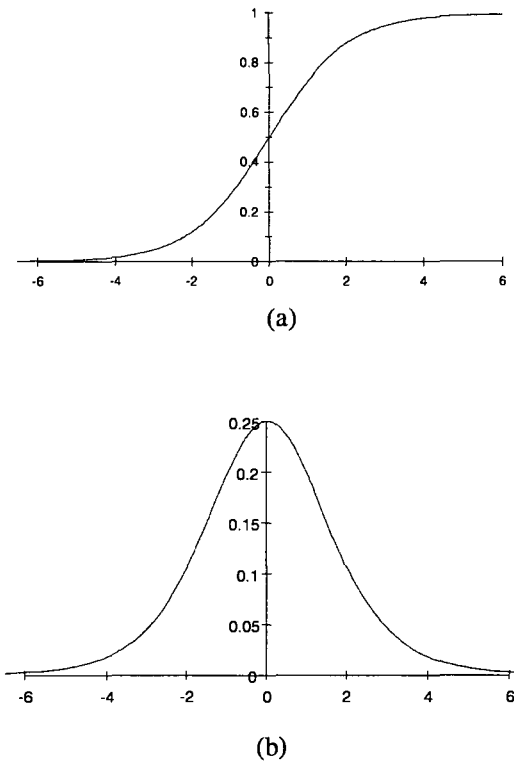

Fig. 3.9 Graphical representation of (a) Logistic function, and (b) Derivative of logistic function

For hidden layer neurons, the associated input and output weighted connections are shown in Figure 3.10 to illustrate the weights that are to be considered in the weight update for hidden layer neurons.

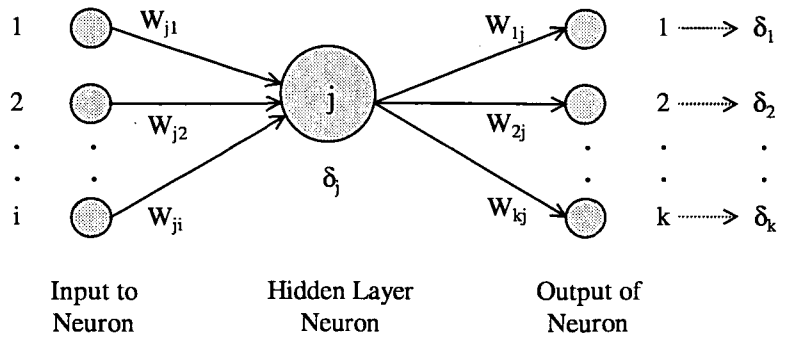

Fig. 3.10 Hidden layer processing neuron [85]

For neuron *j* in the hidden layer the error value calculation considers the weighted sum of the δ values of all neurons that receive output from neuron *j*. Hence, the error value calculation for the hidden layer is written as [85]:

$$\delta_j = \left[\sum_k \delta_k w_{kj}\right] f'(\text{net}_j) \tag{3.18}$$

where, w_{ki} = weight connection to neuron *k* from neuron *j*

The weight adjustment for the interconnections of the output and hidden layer neurons is now calculated using the respective δ values. Therefore, each interconnection weight is adjusted by considering the δ value of the neuron that receives input from that interconnection. Hence, the interconnection weight adjustment is written as [85]:

$$\Delta w_{ji} = \eta\, \delta_j\, a_i \tag{3.19}$$

where, w_{ji} = weight connection to neuron *j* from neuron *i*, and

η = learning rate constant, $0 < \eta < 1$

It can be seen from Equation 3.7 that updating of interconnection weights is primarily based on three parameters, η, δ_j and a_i. As Δw_{ji} is proportional to δ_j then it follows that a large error value from neuron *j* will result in a large adjustment to its incoming

weights. Similarly, large output values, a_i, will result in larger weight adjustments. The learning rate, η, in the weight adjustment equation is selected to reflect the desired convergence speed of the neural network. However, very large values of η lead to instability in the network and unsatisfactory learning while very small values of η give rise to an excessively slow learning rate. Sometimes the learning rate is varied in an attempt to produce a more efficient learning rate for the network; for example, allowing the value of η to begin at a high value and to decrease during the learning session can sometimes produce better learning performance [85].

One of the most popular ways to improve the convergence of the weight update is the introduction of a momentum term. The weight adjustment at each iteration is shown in the following equation and is referred to as the 'generalised delta rule' [86]:

$$\Delta w_{ji}^{k} = \eta \, \delta_j \, a_i + \alpha \Delta w_{ji}^{k-1} \tag{3.20}$$

where, α = momentum term constant, $0 \leq \alpha < 1$

Δw_{ji}^{k} = weight update at iteration *k*

Δw_{ji}^{k-1} = weight update at iteration *(k-1)*

Through addition of the momentum term a portion of the weight update is not applied until the following iteration, resulting in dampening of oscillations in weight changes and improved convergence [110].

The weights of the network to be trained are typically initialised at small random values [67]. A well known initialisation heuristic for a feedforward network with sigmoidal units is to select its weights with uniform probability from an interval $(-\alpha, \alpha)$ [104]. A common procedure is to initialise weights to random values between -0.5 and 0.5, or between -1 and 1, or some other suitable interval [109]. Weight initialisation is an important procedure as it has substantial influence over network convergence. If all weights start out with equal values, and if the solution requires that unequal weight values be developed, the network may not train properly. Unless random factors, or the random character of input patterns during training, disturb the network, the internal representation may continuously result in symmetric weights. In

particular, because the network weight update procedure incorporates the difference between a neuron output and the target output, if each neuron output within the network is the same value, due to equal valued weights, each weight change within the network will be identical and hence, the network weights will never differ. This is not desirable however, as most applications require uneven weights within the network, hence, a network with even weights will not produce accurate results.

3.5.2 Algorithm

The training procedure for a backpropagation network is written as follows [75]:

Step 1. Initialise the weights of the network at small random values.

Step 2. Start the learning cycle by exposing the network to a certain input pattern paired with the desired output.

Step 3. Compute the network's output, Equations 3.15 and 3.16, and compare it with the desired output so that the error can be calculated, Equations 3.17 and 3.18 for the output and hidden layers, respectively.

Step 4. Adjust the weights of the network using the error backpropagation algorithm so that a certain amount of the detected error is removed, Equation 3.19, or 3.20 if a momentum term is used.

Step 5. Repeat Steps 2 to 4 with all the input patterns and their correspondent desired outputs (training examples). Compute the cumulative error, Equation 3.14.

Step 6. If the cumulative error is within a tolerable range, terminate the training process, otherwise, go back to Step 2.

3.5.3 Applications

Backpropagation network models have gained significance as an important technique to be applied for various practical applications [76, 83]. Backpropagation networks can be applied to almost all applications in the manufacturing domain and in fact, they are the most popular neural network models in manufacturing applications [75]. Backpropagation networks are applied to various aspects of manufacturing engineering such as design applications, scheduling, monitoring, diagnosis and quality assurance [79-82, 84].

3.6 RADIAL BASIS FUNCTION NETWORK

Neural networks based on localised basis functions and iterative function approximation are usually referred to as radial basis function, RBF, networks [135]. An RBF network is a type of feedforward neural network that learns using a supervised training technique. Broomhead and Lowe [118] were the first to exploit the use of radial basis functions in the design of neural networks. Other major contributions to the theory, design and application of RBF networks include work by Moody and Darkin [119], Renals [120] and Poggio and Gorosi [121]. Radial functions are a special class of function, their characteristic feature is that their response decreases, or increases, monotonically with distance from a central point [129]. The centre, the distance scale and the precise shape of the radial function are parameters of the model [129]. It has been shown that RBF networks are able to approximate any reasonable continuous function mapping arbitrarily well [119, 120, 121, 130] and with the best approximation property [131].

3.6.1 Architecture

An RBF network, in its most basic form, is comprised of three different layers; an input layer, a hidden layer and an output layer, as shown in Figure 3.11. There is also typically a bias unit on each output node. The primary adjustable parameters are the final layer weights, w_{kj}, connecting the *kth* output node to the *jth* hidden layer node [133]. There are also weights connecting the input nodes to the hidden layer nodes. All hidden layer nodes are connected to all input nodes and all output nodes. However, there are no connections between non-adjacent layers.

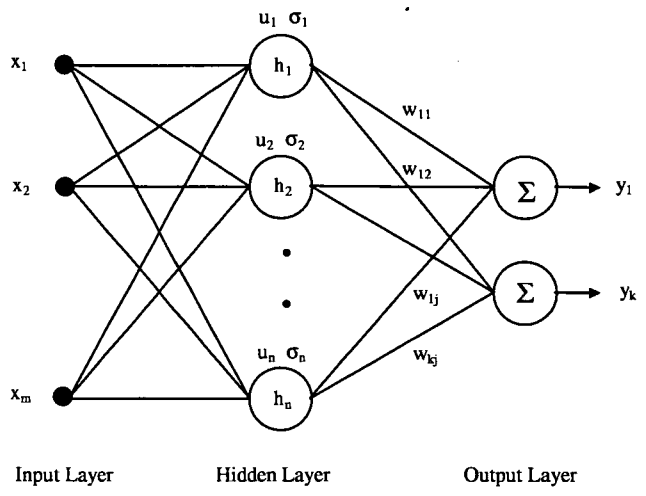

Fig. 3.11 Basic architecture of a radial basis function network [117]

The function of the input layer is to simply pass forward to all hidden layer neurons the activation patterns applied to the network, hence, it is a linear process. In addition, the purpose of the output layer is to supply the response of the network to the activation patterns applied to the input layer, commonly using a linear function. Generally, a linear weighted summation of the following form is used [79]:

$$y = \sum_i h_i w_i \tag{3.21}$$

where, h_i = output of hidden layer, and

w_i = output layer weight

The number of input and output nodes are determined by, and equal to, the number of input and output variables respectively in the process to be modelled. That is, the number of input and output nodes are equal to the dimensionality of the input and output vectors respectively. However, the hidden layer neurons perform a nonlinear computation and the determination of the number of hidden layer neurons is more complex.

The activation function used in the hidden layer neurons of RBF networks is nonlinear, and typically, the Gaussian function is used. This function is of an exponential form, as shown in the following equation [74]:

$$h_n = \exp\left[-\frac{(x-u_i)^T(x-u_i)}{2\sigma^2}\right] \tag{3.22}$$

where, h_n = output of hidden layer neuron *n*,

x = input vector,

u_n = weight vector of hidden layer neuron *n*,

T = indicates the vector transpose, and

σ_i = specifies diameter of receptive field of hidden layer neuron *n*

The shape of the Gaussian function is shown in Figure 3.12, for u = 0 and σ = 1. It can be seen that the function monotonically decreases with distance from the central point.

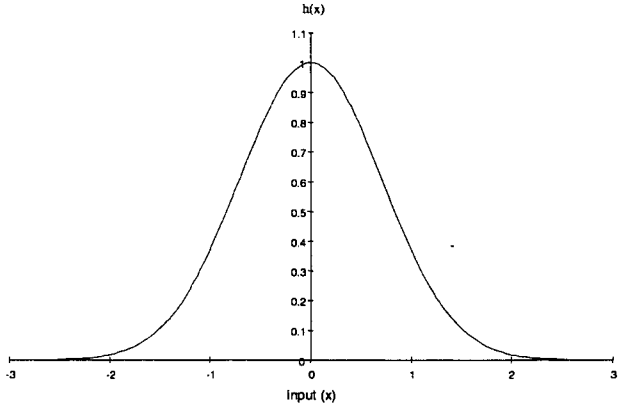

Fig. 3.12 Shape of the Gaussian function using u = 0 and σ = 1

An RBF network has two distinct operating modes, namely, training and reference. During training, the adjustable parameters of the network, u_i, σ_i, and the output layer weight matrix, W, are set so as to minimise the average error between the actual network output and the desired output over the vectors in the training set. Similarly to backpropagation, training in an RBF network involves the reduction of an error. Hence, RMS error is a useful measure of how well the network has learned the

training data. In the reference phase, input vectors are applied and output vectors are produced by the network [79].

Training an RBF network involves assigning values for the function centres, u, and the width of the receptive field, σ, for each hidden layer neuron. The location of the centres of the receptive fields is a critical issue and there are many alternatives for their determination [79]. One option is to have a centre and corresponding hidden layer neuron for each input vector in the training set. However, this can lead to large number of hidden layer neurons, and as a result, computation time is substantially lengthy. An alternative is to locate a hidden layer neuron for a particular cluster of input vectors, significantly reducing the number of hidden layer neurons required and the associated computation time. There are many unsupervised clustering techniques that can be employed to position the centres of the RBF network, and adapt the parameters corresponding to the first layer of the network [134]. A useful clustering technique that can be used for reducing the number of hidden nodes required is given in the following algorithm [74]:

Step 1. Select an input vector x_k from the training set.
Step 2. Find the cluster centre that is closest.
Step 3. Adjust the position of this centre as follows:

$$u_i(n + 1) = u_i(n) + \eta * (x_k - u_i) \tag{3.23}$$

where, n = number of training vectors, and
η = clustering coefficient

Step 4. Repeat Steps 1 to 3 over all input vectors, gradually reducing until changes in u_i are negligible.

Alternatively, in an effort to reduce the number of function centres, the easiest and least computational intensive method is that of random selection; selecting *m* data points randomly from the *n* training data, $m \leq n$, and using the *m* data points as function centres [132]. This method assumes that if the *n* training data patterns form a representative data set, then the randomly selected *m* data points should also be

reasonably representative. However, if the m data points are not representative then the network gives poor approximation as the function centres do not cover the input range. In choosing the number of basis functions to use, a cross validation technique can be adopted. This involves producing a graph of associated training and test set errors with increasing basis functions, Figure 3.13. When the error becomes constant, that is, the slope of the graph becomes and remains horizontal with increasing basis functions, the minimum number of basis functions to use is the point where the slope of the graph starts to become horizontal. Considering Figure 3.13, the optimum number of function centres to use would be 30, as RMS error does not decrease for increasing function centres beyond 30.

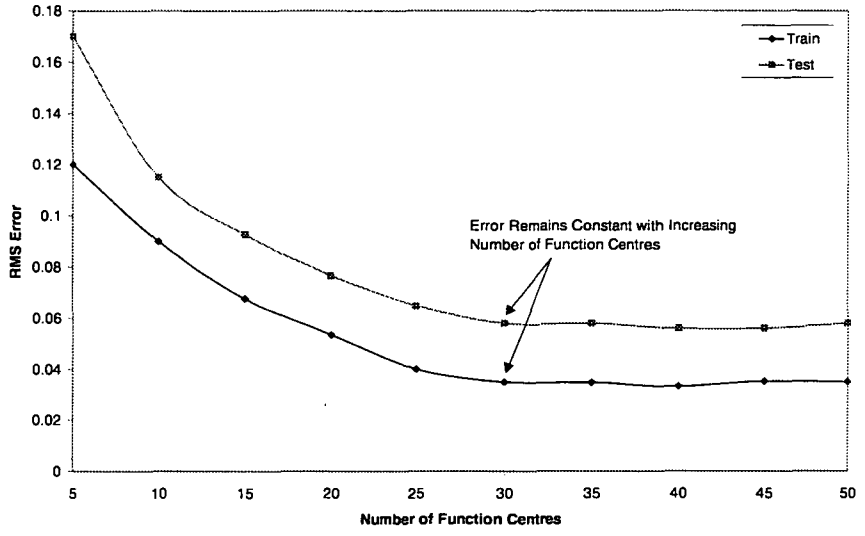

Fig. 3.13 Illustrative graph of RMS error behavior with increasing number of function centres

Alternatively, clustering techniques, such as using a Kohonen or similar type network to cluster the training patterns suitably, can be adopted to reduce the required number of function centres.

The diameter of the receptive field, σ, can have a profound effect upon the accuracy of the system. For hidden layer neurons whose centres are widely separated from others, σ must be large enough to cover the gap, whereas, those in the centre of a cluster must have a small σ if the shape of the cluster is to be represented accurately [79]. In effect, the standard deviation of all the Gaussian radial basis functions is fixed at [115]:

$$\sigma = \frac{d}{\sqrt{2M}} \tag{3.24}$$

where, d = maximum distance between chosen centres, and
M = number of centres

The weight matrix, W, is optimised using a supervised training technique, the generalised delta rule specified for the backpropagation network in the previous section is applicable. Hence, a training set, comprising input vectors and corresponding output vectors, is required for training an RBF network. In RBF network models, the network architecture is determined by the number of nodes in each layer and the location of the function centres [132].

Since the weighting function of the basis function is radially symmetrical, it is usually desirable to normalise the training and test data sets so that each dimension has the same variance [74]. This procedure yields a bounded range of 0 to 1 for the inputs, hence removing any differences in magnitude present among the elements of the input vector.

3.6.2 Algorithm

The training procedure for a radial basis function network is written as follows [75]:

Step 1. Use a suitable clustering technique to set the input-to-hidden layer weights of the network to represent sufficiently the training patterns.
Initialise the hidden-to-output layer weights of the network at small random values.

Step 2. Start the learning cycle by exposing the network to a certain input pattern paired with the desired output.

Step 3. Compute the network's output, Equations 3.21 and 3.22, and compare it with the desired output so that the error can be calculated.

Step 4. Adjust the hidden-to-output layer weights only of the network using the error

backpropagation algorithm so that a certain amount of the detected error is removed, Equation 3.19, or Equation 3.20 if a momentum term is used. The input-to-hidden layer weights remain unchanged from their initial values.

Step 5. Repeat Steps 2 to 4 with all the input patterns and their correspondent desired outputs (training examples). Compute the cumulative error, Equation 3.14.

Step 6. If the cumulative error is within a tolerable range, terminate the training process, otherwise, go back to Step 2.

Moreover, the learning process undertaken by an RBF network can be visualised as follows [115]:

> "The linear weights associated with the output unit of the network tend to evolve on a different time scale compared to the nonlinear activation functions of the hidden units. Thus, as the hidden layer's activation functions evolve slowly in accordance with some nonlinear optimisation strategy, the output layer's weights adjust themselves rapidly through a linear optimisation strategy."

As the different layers of an RBF network perform different tasks, it is reasonable to separate the optimisation of the hidden and output layers of the network by using different techniques, and perhaps operating on different time scales [116].

3.6.3 Applications

Radial basis function techniques are powerful methods with a definite range of applicability [79]. The advantage of this particular type of network is in practical application, the basis of its simplicity is that it combines a linear dependence on the variable weights with an ability to model explicitly nonlinear relationships [118]. RBF networks have been applied to a wide variety of problems, including; image processing [122, 123], speech recognition [124, 125], time series analysis [126, 127] and medical diagnosis [128].

3.7 KOHONEN NETWORK

The principal goal of the Kohonen network, also self-organising feature map or topology preserving map, is to transform an incoming signal pattern of arbitrary dimension into a one- or two-dimensional discrete map, and to perform this transformation adaptively in a topological ordered fashion [115]. Teuvo Kohonen [90-94] has been the primary developer of this particular type of network. The Kohonen network uses a sort of competition, called lateral inhibition, to ensure that, when implemented, only the correct neurons in the network become active [95]. Competitive learning is based on the notion that elements in the network must compete among themselves for the privilege of modifying their weights [95]. During training, only the 'winning' neuron and its neighbours within a specified physical distance update the weights on their connections, while the remaining neurons undergo no changes to their connections. The winning neuron at time t is the neuron during training that has a weight vector most closely resembling the input vector given at time t. For each input pattern presented to the network during training, a winning neuron is determined and the corresponding weight vector for that neuron is updated. In addition, for a specified number of neurons in the proximity of the winning neuron, those neurons also undergo a weight update. For a sufficient number of iterations, the network weights will converge to a specific set of values. The number of output neurons, or clusters, for the Kohonen network is specified by the number of categories that the input patterns are to placed into. Hence, subsequent to convergence of the weight matrix, the weight vector for a cluster unit serves as an exemplar of the input patterns associated with that cluster.

A significant difference exists between this type of network and the conventional supervised training models in that the correct output cannot be defined *a priori*, hence, a numerical measure of the magnitude of the mapping error is not possible. However, the learning process leads to the determination of well-defined network parameters for a given application [104].

3.7.1 Architecture

The architecture of the basic Kohonen network consists of only an input layer and an output layer, shown in Figure 3.14. There are i input units and k output units, with all input units being connected to all output units through a weighted connection, w_{ki}.

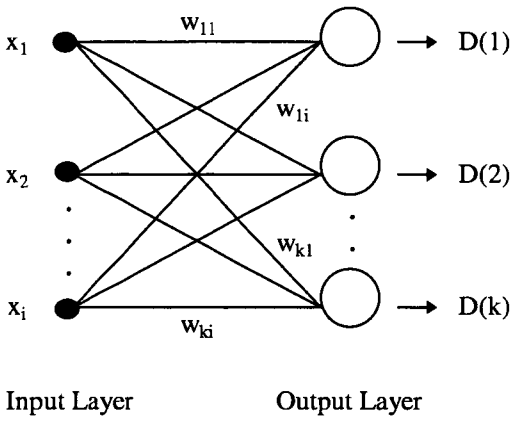

Fig. 3.14 Basic architecture of Kohonen self-organising feature map

For each real output value from the Kohonen network, *D(1)* to *D(k)*, there is a minimum value such that the output neuron with the minimum value is the winning neuron. Only that neuron is then allowed to update its associated weights. This is shown in the following algorithm.

3.7.2 Algorithm

A training algorithm for the Kohonen network is as follows [109]:

Step 1. Initialise network weights, w_{ki}.

Set topological neighbourhood parameter, R.

Set learning rate parameter, η.

Step 2. While stopping condition is false, do Step 3.

Step 3. For each input vector x, do Steps 4 to 9.

Step 4. For each output neuron k, compute:

$$D(k) = \sum_i (w_{ki} - x_i)^2 \quad (3.25)$$

Step 5. Find index K such that D(K) is a minimum.

Step 6. For all output units k within a specified neighbourhood of K, and for all input units i, compute:

$$w_{ki}(new) = w_{ki}(old) + \alpha[x_i - w_{ki}(old)] \quad (3.26)$$

Step 7. Update the learning rate and topological neighbourhood parameters.

Step 8. Reduce radius of topological neighbourhood at specified times.

Step 9. Test stopping condition.

The learning process involved in the computation of a feature map is stochastic in nature, which means that the accuracy of the map depends on the number of iterations of the self-organising feature map algorithm [115]. Furthermore, the success of map formation is critically dependant on how the main parameters of the algorithm, namely, the learning-rate parameter, η, and the topological neighbourhood parameter, R, are selected [115]. The learning rate parameter, η, used to update the synaptic weight vector should be time varying [115]. The learning rate parameter should initially start at a value close to unity, then decrease gradually over the period of iterations to a value above 0.1. The exact form of variation of η is not critical, it can be linear, exponential or inversely proportional to the number of iterations [115]. For topological ordering of the weight vectors to take place, careful consideration has to be given to the neighbourhood parameter, R. Initially, the topological neighbourhood parameter should be set to a value such that it includes all neurons in the network, then gradually decreases with increasing iterations such that towards the final iterations the topological neighbourhood parameter includes only the winning neuron, and maybe 1 neighbour.

3.7.3 Applications

Neural networks developed by Kohonen have been applied to an interesting variety of problems, with one application of this type of network to computer-generated

music [109]. Self-organising networks have been implemented successfully for applications including speech recognition [96,97] and tool wear pattern recognition for turning operations [98]. Character recognition using the Kohonen self-organising feature map has also had substantial success, using the network to cluster input patterns representing different letters of the alphabet. In addition, the Kohonen network has also been applied successfully to the well-known travelling salesman problem.

3.8 GENERAL REGRESSION NEURAL NETWORK

The general regression neural network, or GRNN, was discovered by Donald Specht in 1990 [137] and is based on the previously developed Nadaraya-Watson kernel regression [140]. A GRNN is a memory based feedforward neural network, it responds to an input pattern by processing the input data from one layer to the next with no feedback paths. The GRNN is a function approximator system, which is useful for estimating the values of continuous variables such as future position, future values, and multi-variable interpolation [136]. GRNNs feature fast training times, can model nonlinear functions, and have been shown to perform well in noisy environments given enough data [139].

When using the GRNN, if the variables to be estimated are future values, the GRNN is a predictor. If they are dependent variables related to input variables in a process, plant or system, the GRNN can be used to model the process, plant or system [136]. In both cases the GRNN can instantly adapt to new data points in a very short time by including the new data points in the training set.

The primary advantage of the GRNN is the speed at which the network can be trained. Training a GRNN is performed in one pass of the training data through the network, the training data values are copied to become the weight vectors between layers. While the advantages of the GRNN include fast training times, ability to handle both linear and nonlinear data and the fact that the smoothing parameter is the only adjustable parameter, thereby making overtraining less likely, the GRNN also has some associated disadvantages. For example, the GRNN requires many training

samples to adequately span the variation in the data, and it requires that all training samples be stored for future use. In addition, the GRNN has trouble with irrelevant inputs and there is no intuitive method for selecting the optimal smoothing parameter.

3.8.1 Architecture

The architecture of a basic GRNN, shown in Figure 3.15, has four layers; input, pattern, summation and output, with weighted connections w_{ji} between the input and pattern layer and A_i and B_i between the pattern and summation layer. There are i input neurons, j pattern neurons, $k+1$ summation neurons and k output neurons.

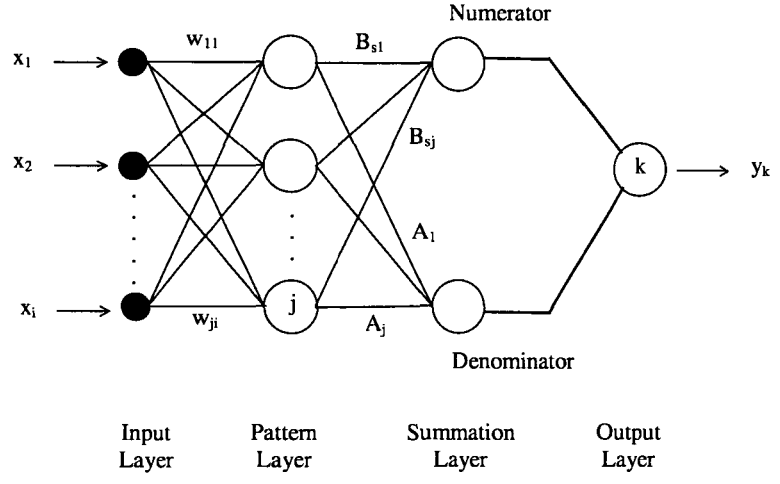

Fig. 3.15 Architecture of basic general regression neural network

The function of the input layer is to pass forward the activity patterns presented to the network to all neurons in the pattern layer. The number of input layer neurons is equal to the dimension of the input vector. The neurons in the pattern layer perform a nonlinear transformation of the input patterns. When a new vector X is entered into the network, it is subtracted from the stored weight vector representing each activity pattern. Either the squares or the absolute values of the differences are summed and fed into a nonlinear activation function [137]. The activation function normally used is the exponential function shown in the following equation:

$$f(net_i) = \exp[\frac{-net_i}{\sigma}] \quad (3.27)$$

where, $f(net_i)$ = output from pattern layer neuron *i*,

net_i = sum of differences between input and weight vector for pattern layer neuron *i*, and

σ = smoothing factor

The output from all neurons in the pattern layer then becomes input for all neurons in the summation layer. For a single output network the summation layer consists of a denominator neuron and a numerator neuron. For each additional output unit a single numerator is added. Hence, the summation layer consists of a single denominator unit and *n* numerator units, where *n* equals the number of output neurons. The summation layer neurons perform a dot product between a weight vector and a vector composed of the signals from the pattern units [137]. For the denominator summation neuron, the weight vector is unity, so a simple sum is performed, represented by the following equation:

$$den_{out} = \sum_j f(net_j) * A_j \quad (3.28)$$

where, den_{out} = output from denominator summation neuron,

$f(net_j)$ = output from pattern layer neuron *j*, and

A_j = weight connecting denominator summation neuron to all pattern layer neurons, equal to one for all A_j

For the numerator summation neuron, the weight connecting it to each pattern layer neuron is equal to the value of the dependent variable for the training case of that pattern layer neuron [138]. Hence, the numerator summation neuron performs a computation represented by the following equation:

$$num_{outs} = \sum_j f(net_j) * B_{sj} \quad (3.29)$$

where, num_{outs} = output from numerator summation neuron *s*,

$f(net_j)$ = output from pattern layer neuron j, and

B_{sj} = weight connecting numerator summation neuron s to all pattern layer neurons

The output from the denominator and numerator summation neurons are sent to the output layer neurons, the function of which is to divide the output of the associated numerator summation neuron by the output of the denominator summation neuron.

3.8.2 Algorithm

As a preprocessing step, it is usually necessary to scale all input variables such that they have approximately the same ranges of variances. The need for this stems from the fact that the underlying probability density function is to be estimated with a kernel that has the same width in each dimension [136].

A training algorithm for the general regression neural network is as follows:

Step 1. Determine a suitable value for the smoothing parameter, σ.

Step 2. Set input to pattern layer weights, w_{ji}, equal to the values of the independent variables in the training data set.

Step 3. Set pattern to summation layer denominator weights, A_j, equal to a value of one.

Step 4. Set pattern to summation layer numerator weights, B_{sj}, equal to the values of the dependent variables in the training data set.

Step 5. Pass the entire training data set through the network and calculate the network output in each instance using Equation 3.27 for the pattern layer neurons, and Equations 3.28 and 3.29 for the summation layer denominator and numerator neurons, respectively.

Step 6. Calculate network output by dividing the numerator output by the denominator output for k output neurons.

Step 7. Compute and observe the prediction error of the network by comparing the predicted output with the target output.

Step 8. If the prediction error is unacceptable, change the value of the smoothing

parameter and repeat Steps 5 to 7.

Step 9. Repeat Step 8 until the prediction error is minimum.

The optimisation of the smoothing factor, σ, is critical to the performance of the GRNN [139]. A useful method for selecting σ is the 'hold out' method, which involves examining the prediction error of the network for different values of σ [136]. The value of σ that produces the lowest error then becomes the selected smoothing factor.

3.8.3 Applications

The general regression neural network is suitable for prediction, modelling, mapping and interpolation, or as a controller. In particular, the fields of nonlinear control systems and robotics are particularly good application areas that can use the potential speed of GRNNs [137]. Narendra and Parthasarathy [141] use a GRNN to separate the problem of control of nonlinear dynamical systems into an identification or system modelling section, and a model reference adaptive control section. In the identification model, a GRNN was used to approximate the function representing the system behaviour.

3.9 CONCLUDING REMARKS

It is evident that for online monitoring a neural network system is required It is an important decision however as to which particular model is chosen. The architecture of the individual models is described in this chapter, based on their performance prediction for a given situation. The next chapter discusses the most reliable neural network model for this field of work.

CHAPTER 4

EXPERIMENTAL OBJECTIVE

The objective of this experiment was to èstablish a reliable online evaluation of the status and conformity of the surface finish of the work material. This was done by first, a direct modeling process, testing various parameters on a standard lathe, through metal turning. In varying speed, feed, depth of cut, and tool geometry parameters, the output components will be forces (two components), frequency, and an offline determination of surface finish. Labview (a graphical data output interface) was set up to measure and record these output variables. Figure 4.1 below shows the direct modeling process.

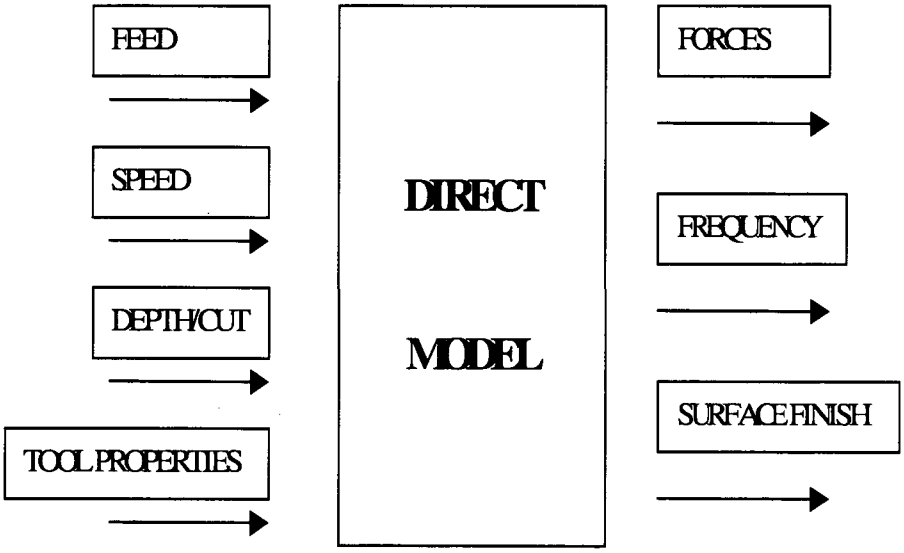

Figure 4.1 Direct model setup

Through this direct model we integrate the use of a neural network system (explained in depth in Chapter 3). The neural network system now uses our outputs from our direct model, as inputs for the inverse model shown in figure 4.2.

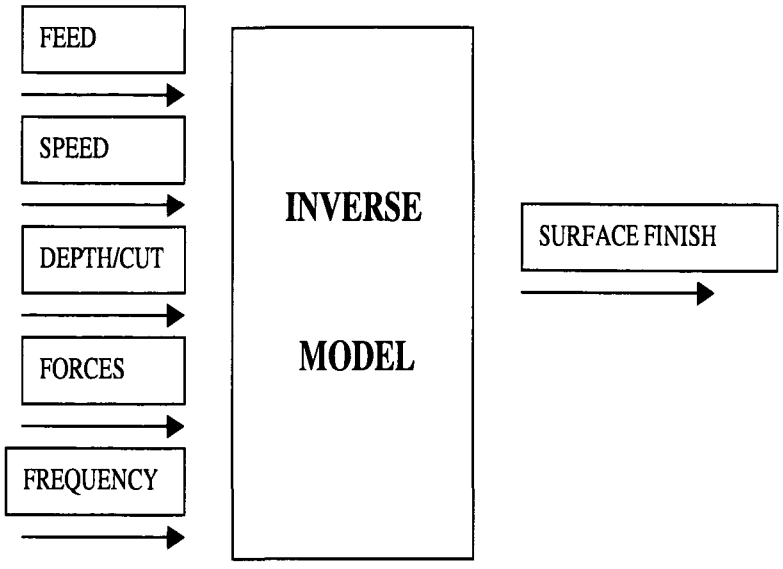

Figure 4.2 Inverse model setup

This model represents that, given experimental data has been obtained, a neural network can be trained with that data, and then using the neural network analysis, the expected surface finish data can become an output while the system is still online. These neural network output values can then be compared with the experimental offline data, to see what sort of error conformance the entire process introduces.

To this day, surface finish is only determined offline, in what can be a costly and time consuming practice. For example, in automobile wheel manufacturing, quite often based on customer's demands the process needs to be modified. A surface profile of certain surface finish demanded by the customer requires proper selection of tools. In this situation, surface finish becomes input for the right tool selection and geometry (as the outputs). This above technology introduces the ability to automatically detect when work material becomes out of the range of acceptable quality limits, and can, with the aid of present technology, alleviate the problem almost instantly, making the production process almost fully automated. The neural network or inverse model is displayed in figure 4.2 only as a summary of the overall implementation process. Figure 4.3 displays the entire inverse modeling process.

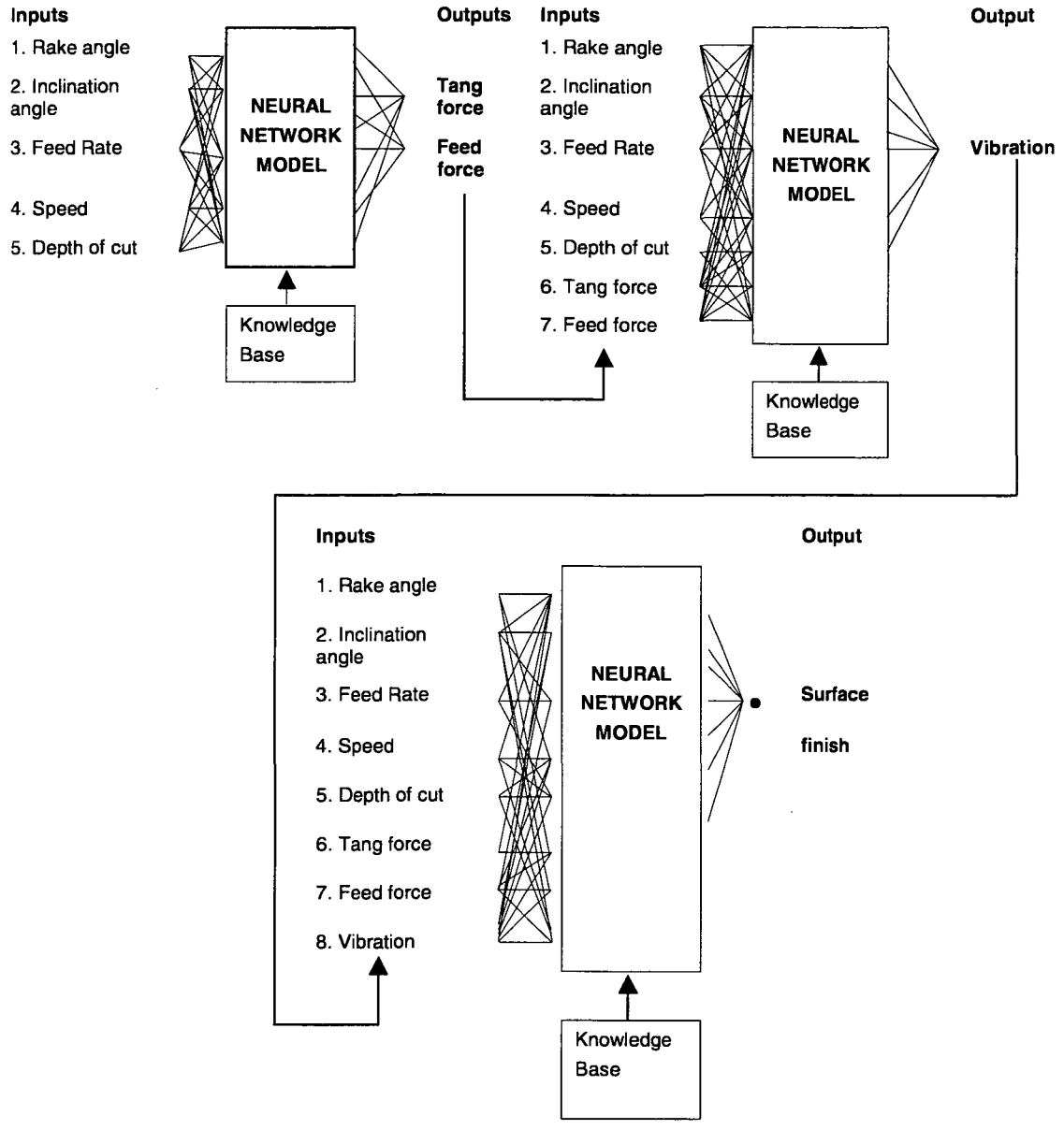

Fig.4.3 Cascade type neural network technique for performance prediction

Diagram 4.3 tells us that after the input variables have been obtained (through extensive testing), the neural network model outputs its prediction values of the two force components. Given that the data acquisition has already obtained experimental values for these force components, a comparison can be made. Assuming that the training and testing data is accurate, an error analysis will be carried out on the trend data. If the error range is small (for example +/-5%), then the next neural stage will begin.

The two force components will now become inputs for the next stage. The output in this case will be the frequency of vibration. Similarly the values will be compared with experimental values, and if a good error acceptance is given, then the final model will begin. This is the crux of the project. Now all the variables (other than surface finish) become inputs, and the neural network model now predicts surface finish values. Given that the input variables are within a reasonable error margin, then the output surface finish values are speculated to be reasonably accurate and compare well with experimental values. Given this procedure occurs then the testing of a workpiece materials surface finish can be enabled online.

For the indirect application of this project, a neural network model will be used (discussed previously). In particular, a Back Propagation Neural Network (BPNN) will be used. This type of neural network is discussed in greater depth in the previous chapter. Backpropagation is the most widely used of the neural network paradigms and has been applied successfully in applications studies in a broad range of areas [78, 99, 100, 101]. Backpropagation networks are multi-layered feedforward neural networks (explained extensively in chapter 3) that are trained using the error backpropagation procedure, a supervised mode of training. Backpropagation is a systematic method for training multi-layered artificial neural networks and is a form of supervised training [71].

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (EW). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the EW.

The back-propagation algorithm is easiest to understand if all the units in the network are linear. The algorithm computes each EW by first computing the EA, the rate at which the error changes as the activity level of a unit is changed. For output units, the EA is simply the difference between the actual and the desired output. To compute the EA for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is

connected. We then multiply those weights by the EAs of those output units and add the products. This sum equals the EA for the chosen hidden unit. After calculating all the EAs in the hidden layer just before the output layer, we can compute in like fashion the EAs for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the EA has been computed for a unit, it is straight forward to compute the EW for each incoming connection of the unit. The EW is the product of the EA and the activity through the incoming connection.

Note that for non-linear units, the back-propagation algorithm includes an extra step. Before back-propagating, the EA must be converted into the EI, the rate at which the error changes as the total input received by a unit is changed.

The architecture of a backpropagation network, shown in Figure 3.8, consists of an input layer, one or more hidden layers and an output layer. There are *i* input nodes, *j* hidden nodes and *k* output nodes. All input nodes are connected to all hidden nodes through weighted connections, w_{ji}, and all hidden nodes are connected to all output nodes through weighted connections, w_{kj}.

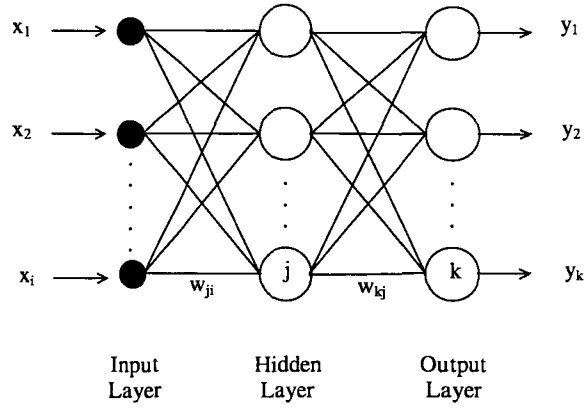

Fig. 4.4 Basic Structure of a Backpropagation Network [66]

The sole function of the input neurons is to pass forward input patterns to neurons in the hidden layer. In this type of feedforward network there are no connections leading from a unit to units in previous layers, nor to other units in the same layer, nor to units more than one layer ahead [66]. Hence, every neuron in each layer

communicates only with neurons in the immediate following layer. Only the hidden layer and output layer neurons complete any type of processing. The neurons in these layers perform three functions; an input function, an activation function and an output function. The input function performs a summing of the inputs and synaptic weights. Hence, the input function summation is used in the processing performed by activation function. The purpose of the output function is to pass forward the output of the activation function. Hence, the output function is a linear function, it is equal to the output of the activation function.

For hidden layer neurons, the associated input and output weighted connections are shown in Figure 3.10 to illustrate the weights that are to be considered in the weight update for hidden layer neurons.

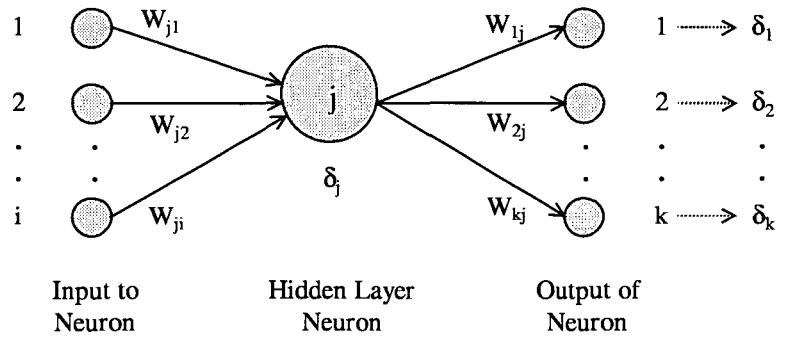

Fig. 4.5 Hidden Layer Processing Neuron [85]

The weights of the network to be trained are typically initialised at small random values [67]. A well known initialisation heuristic for a feedforward network with sigmoidal units is to select its weights with uniform probability from an interval ($-\alpha$, α) [104]. A common procedure is to initialise weights to random values between -0.5 and 0.5, or between -1 and 1, or some other suitable interval [109]. Weight initialisation is an important procedure as it has substantial influence over network convergence. If all weights start out with equal values, and if the solution requires that unequal weight values be developed, the network may not train properly. Unless random factors, or the random character of input patterns during training, disturb the network, the internal representation may continuously result in symmetric weights.

This BPNN needs comprehensive experimental data and design/development of the test rig. For training and testing purposes the test processing is selected from an experimental matrix (see chapter 5). This particular test setup required to comply with the experimental, matrix is discussed in depth in the next chapter.

CHAPTER 5

DEVELOPMENT OF EXPERIMENTAL TEST RIG:

The previous chapter mentioned the guidelines for this project. For the BPNN prediction to be a quantitative model, there is a need for extensive training so that the model is capable of predicting the process variable outcomes accurately. Therefore both the direct and inverse modeling techniques need to be carried out with extensive precision to ensure an accurate database of information.

The direct modeling technique initially requires certain process variables (inputs), and the corresponding performance variables (outputs). For this particular experiment we have (table 5.1):

Process Variables	Performance variables
Speed (V) Feed (f) Depth of cut (d) Tool Geometry (α,I) Tool material Tool hardness Work Material	Tangential Force (F_{Tang}) Feed Force (F_{Feed}) Radial force = 0 (orthogonal cutting) Frequency of vibration (f_v) Surface finish (R_a, R_y)

Table 5.1 Direct variables

This chapter deals with the set up of the experimentation required to achieve these performance variables.

5.1 EXPERIMENTAL TEST RIG

To discuss the basic experimental setup, each component will be discussed individually, based on the setup diagram (figure 5.1). The complete setup gives, after each individual cut, values for the performance variables, given the appropriate process variables are in place. Given the repetition of this cutting and recording process, the adequate data will be obtained for the inverse neural network model.

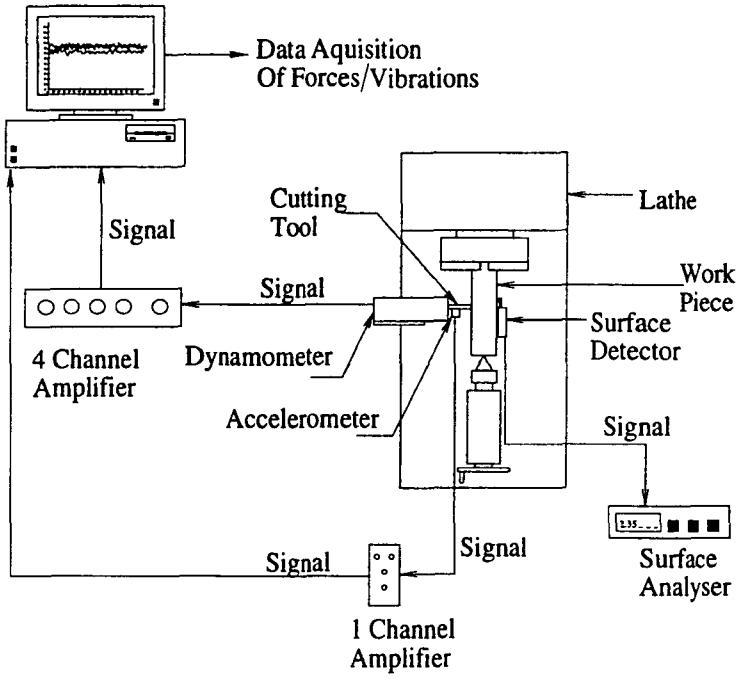

Figure 5.1 Experimental test setup

Each of these individual components will now be discussed individually.

5.1.1 The Lathe

The experiments were carried out on a Harrison VS330TR, 330mm (13inch) swing variable speed center lathe. The Harrison lathe is a high precision manual lathe with a variable speed controller. The lathe used for experimentation is shown below in figure 5.2:

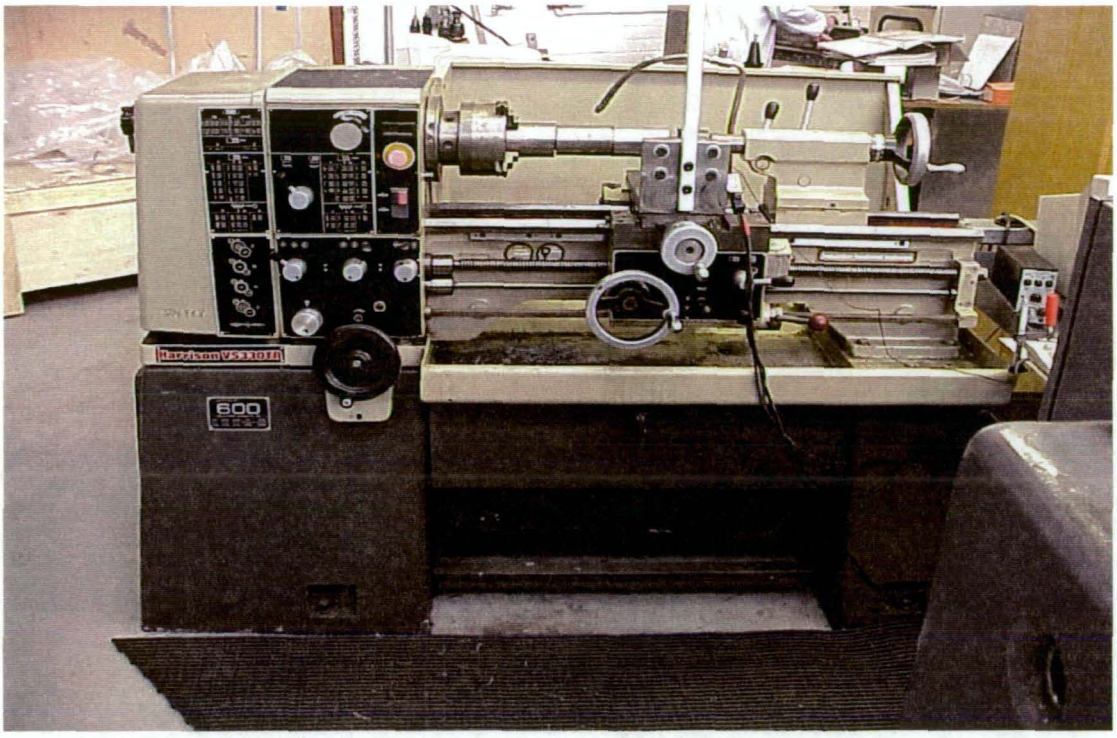

Figure 5.2 Harrison VS330TR Lathe

The variable speed controller enables the spindle to rotate anywhere between 35 rpm – 3000 rpm as desired. The speed is varied manually through the controller shown in figure 5.3. The reason why this is discussed is due to the experimental matrix (Appendix C). The lathe has the characteristics displayed in table 5.2.

SPECIFICATION	CATEGORY	VALUE
Centers	Height	167mm
	Admits between	635mm
Swing	Over bed	330mm
	Over cross slide	210mm
	In gap diameter	480mm
	In gap length	115mm
Spindle	Bored to pass	38 mm diameter
	Nose	No. 4-d1 Camlock
	Morse taper in nose	No. 5
	Morse taper in bush	No. 3
Speeds	Infinitely variable	
	Ranging from	35 – 3000 rpm
Motor	1000rpm @ 50 Hz	2.2 kW 3hp

Lead screw	Diameter	28mm
	Thread	6mm pitch or 4 TPI
Threads	39 Metric pitches	From 0.2 to 14mm pitch
	35 English pitches	From 2 to 56 TPI
	18 Module pitches	From 0.3 to 3.5 MOD
	18 Diametral pitches	From 8to 56 DP
Feeds	16 Metric (R.10 Series)	From 0.03 to 1mm/rev
	16 English	From 0.0001 to 0.040 inch/rev
Cross slide	Width	140mm
	Travel	190mm
Top Slide	Width	82mm
	Travel	92mm
Tool	Max section	16 * 20mm
Tailstock	Quill Diameter	42mm
	Quill travel	110mm
	Quill morse taper	No. 3
	Set-over	+/- 12mm
Weight	630mm Model	610kg
	630mm Cts	

Table 5.2 Lathe specifications

The experimental matrix requires steps in speeds between 0 and 1000 rpm. Therefore there is a constant need for use of the manual speed control. To simply change speed, the dial is set to the speed range, and then fine tuned with another manual control until the desired speed is achieve

The lathe is also highly accurate in its feed settings. It allows for the operation of feed in the range of 0.03mm/revolution – 1mm/revolution. The testing matrix requires only low feed rates, from 0.03 mm/revolution - 0.25 mm/revolution. The feed is also manually controlled, but also requires you to change gear configurations in order to achieve your required feeds. Figure 5.3 and figure 5.4 show the manual configuration setting.

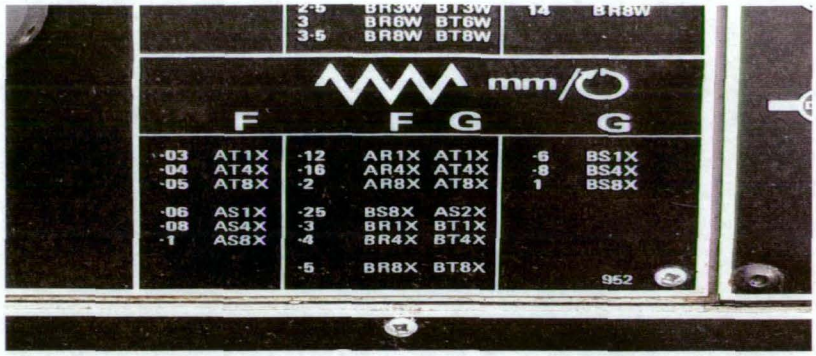

Figure 5.3 Manual feed settings

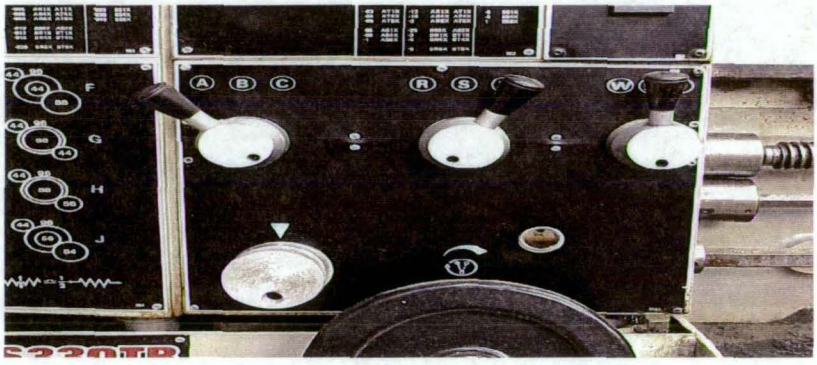

Figure 5.4 Manual feed gearing

Now say for instance the testing matrix requires a feed of 0.16mm/revolution, then the gauges would need to be set for AR4X (as shown in figure 5.4). This is achieved by setting the first lever to A, the second to R, the third to X, and the dial in the bottom left hand corner, to 4, as shown in figure 5.4. The depth of cut is achieved by moving the manual controller as shown in figure 5.5.

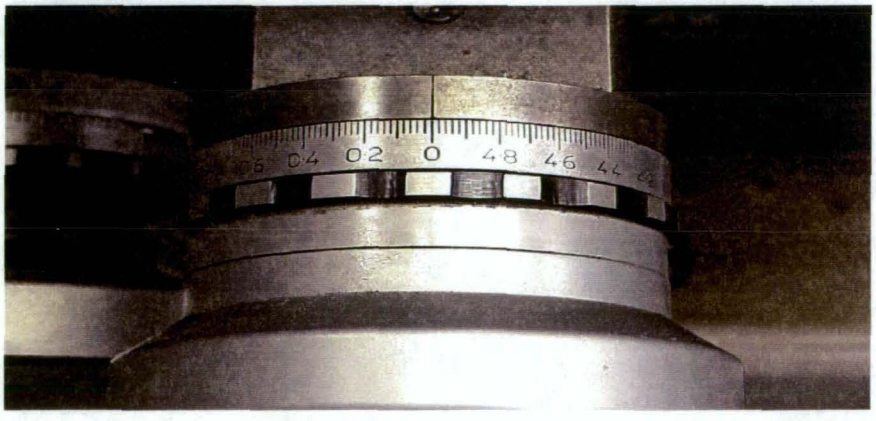

Figure 5.5 Manual depth of cut controller

The lathe allows for 100^{th} of centimeter accuracy when determining the amount of material to remove. The gauge in figure 5.5 represents the depth of removal on the diameter of the work piece.

5.1.2 Cutting Tools

The cutting tools used for this experimentation are of type Sandvik Coromant. They are of the mild steel CNMG 4025 series, and include a PF grade, and a PM grade, as shown below in figure 5.6.

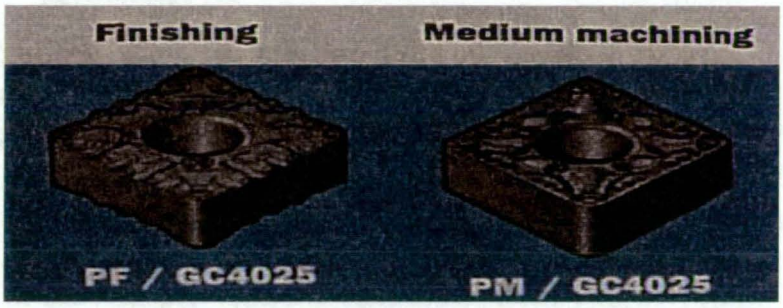

Figure 5.6 Tool types

The inserts are for standard steel cutting and for average to difficult machining conditions. Two types of tools have been chosen.

1. PF. A tool devoted to low machining, or finishing. Slightly softer and a lower incidence angle than the other type of tool

2. PM. A medium machining tool, for mid range conditions.

The geometry of the tool inserts is demonstrated in figure 5.7.

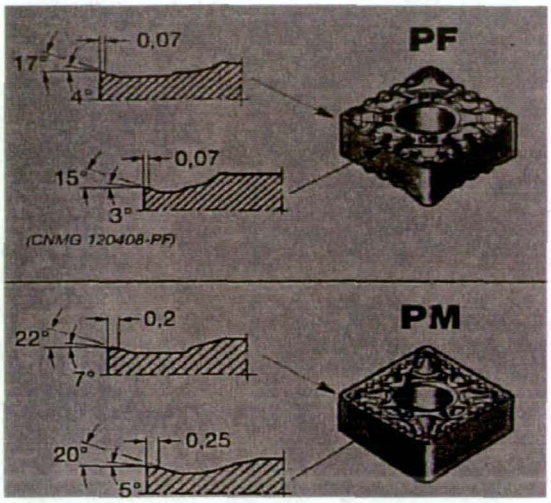

Figure 5.7 Individual tool geometry

These geometry's specified in figure 5.7, become variables in the testing matrix (appendix 1), for which prove an important part of the direct modeling process.

The tool holder is a standard MVC 003, used for the above type of inserts. Diagram 5.8 shows the tool holder, and the insert, present in the lathe setup.

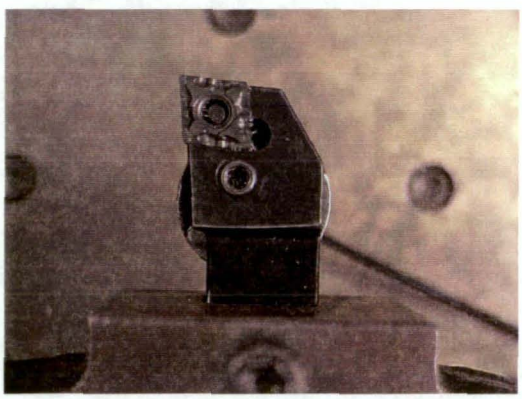

Figure 5.8 Tool insert and holder

5.1.3 Dynamometer

The dynamometer used to measure tangential, feed and radial forces was is a concentric dynamometer designed and built at the University of Melbourne. The dynamometer was mounted on a bracket made specifically to set the tool cutting

point at the center line of the work piece and both the bracket and dynamometer are mounted on the cross slide of the lathe, which can be seen in figure 5.9. The tool holder and insert are mounted in the dynamometer with the tool tip being far enough away from the dynamometer edge for the accelerometer to be mounted.

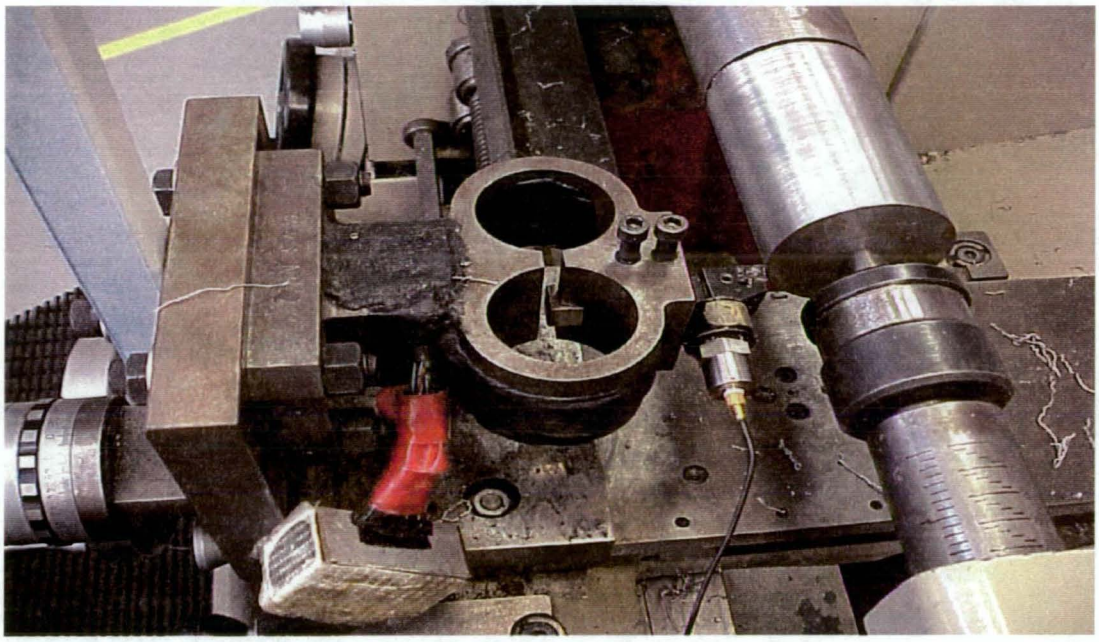

Figure 5.9 Dynamometer and components

The dynamometer was specifically designed for measurement of forces on a single point tool during the turning operation. As seen from figure 5.10, the dynamometer basically consists of two proving rings combined on a mounting shaft and twelve strain gauges for the possible measurement of the three axis forces:

1. Tangential Force.
2. Feed Force.
3. Radial Force.

Strain gauges 1,2,3,4 measure the tangential force, gauges 5,6,7,8 measure the feed force, and strain gauges 9,10,11,12 measure the radial force in turning operation. Each strain gauge is of the bonded resistance type and each set of four strain gauges is set up into a wheatstone bridge, which is a common circuit for measurement of small changes in voltage.

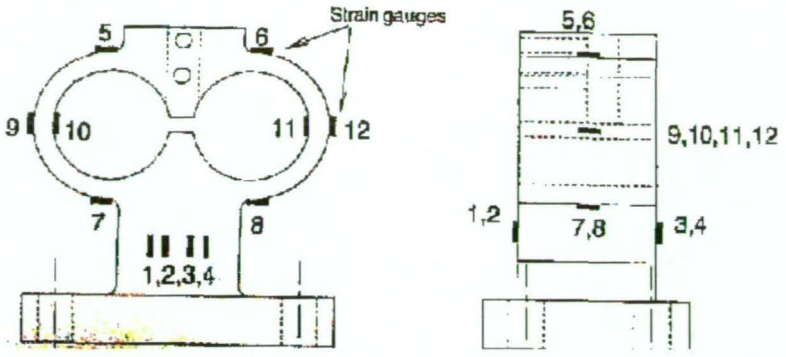

Figure 5.10 Concentric dynamometer with strain gauges

The amplifier used was a four channel continuously exciting amplifier built specifically by the Manufacturing Research group at University of Tasmania. It possesses coarse and fine tuning capabilities for each channel to filter out any noise. The amplifier is represented in figure 5.11.

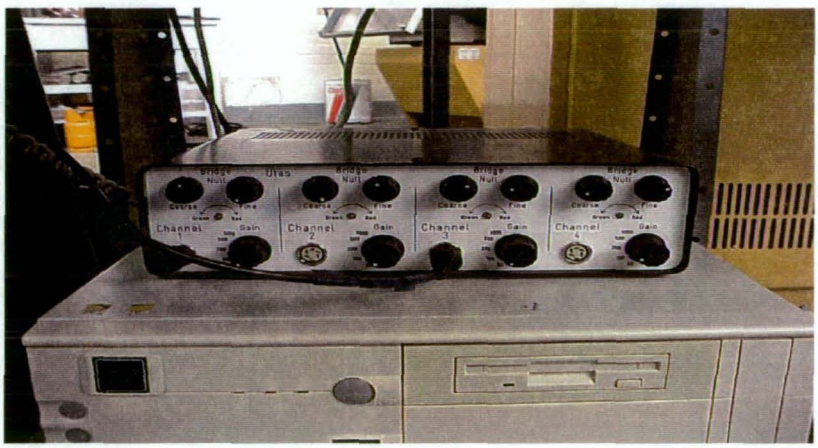

Figure 5.11 Amplifier used for force acquisition

5.1.4 Accelerometer

The accelerometer used to measure frequency of vibration of the work tool was a Bruel & Kjaer type 4368 accelerometer seen in figure 5.12. The accelerometer is of the piezoelectric compression type with physical properties (see appendix A) shown in table 5.3.

PROPERTY	PARAMETER
Weight	30 grams
Material	Titanium
Electrical Connector	Normal coaxial 10-32 thread
Reference sensitivity	At 50Hz at 23 deg C
Cable capacity	114pF
Voltage sensitivity	36.3 mV/g
Charge sensitivity	47.9 pC/g
Capacitance (including cable)	1320 pF
Maximum transverse sensitivity	At 30Hz is 1.5%
Frequency range	10000 Hz

Table 5.3 Accelerometer characteristics

Figure 5.12 shows the position and orientation of the accelerometer and its magnetic base in conjunction with the test setup.

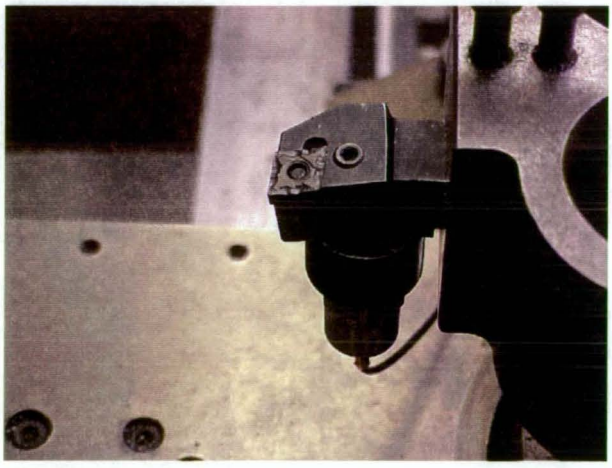

Figure 5.12 The accelerometer in its testing position

The accelerometer signal is conditioned and amplified by a Bruel & Kjaer type 2626 amplifier (see figure 5.13). The amplifier is a multi-purpose charge preamplifier with 3-digit sensitivity conditioning form, selectable lower and upper frequencies plus ground and transformer coupled outputs. The preamplifier is included in the measuring circuit in order to obtain a satisfactory low frequency response.

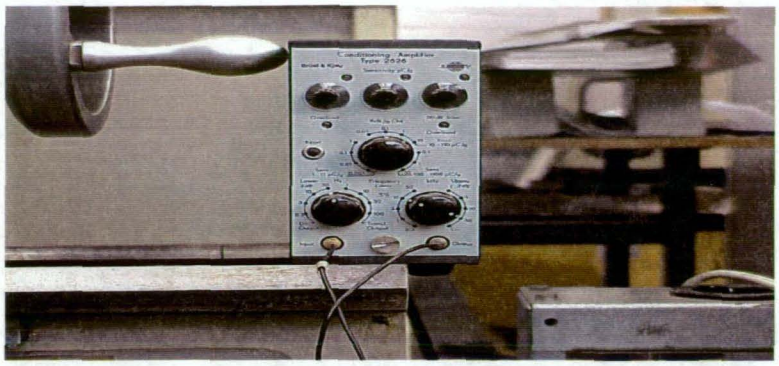

Figure 5.13 Accelerometer amplifier

5.1.5 Surface Analyser

The surface roughness is initially acquired off line using a Mitutoyo Surftest 301 machine (also known as a surface profiler), shown in figure 5.14.

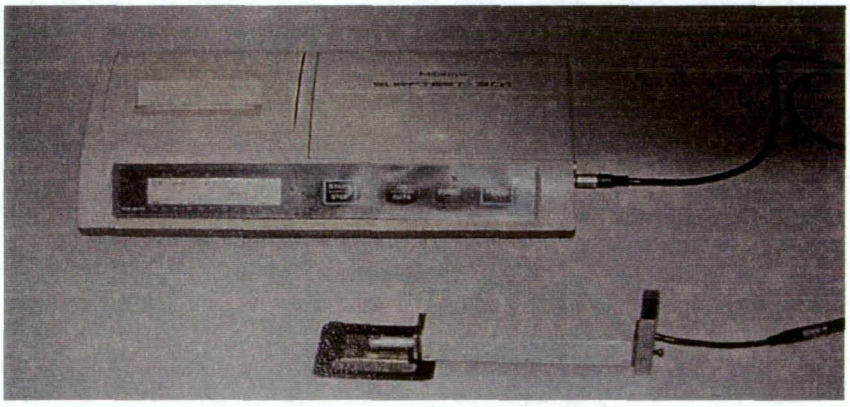

Figure 5.14 Surface profiler

This particular surface profiler is capable of measuring nine different surface profile characteristics:

VARIABLE	DESCRIPTION
R_a	Arithmetic mean deviation of the profile
R_q	RMS deviation of the profile

R_{3z}	Mean peak to valley height
R_t	Maximum peak to valley height
R_y	Same as R_t but with maximum Zi
R_z	Average peak to valley height (max Zi)
R_p	Maximum profile peak height
T_p	Bearing length ratio
P_c	Peak count

Table 5.4 Surface profiler characteristics

For the purpose of this experimentation the most appropriate surface profiles to record are:

1. R_a : With a range of .05 to 40 um
2. R_y : With a range of .3 to 160 um

R_y as the name implies, is the maximum peak to valley height. This is the distance between the highest peak and the lowest valley over the surface of the sample length considered. Figure 5.15 displays the measured components, where the curved surface is the deviation in the output metals surface.

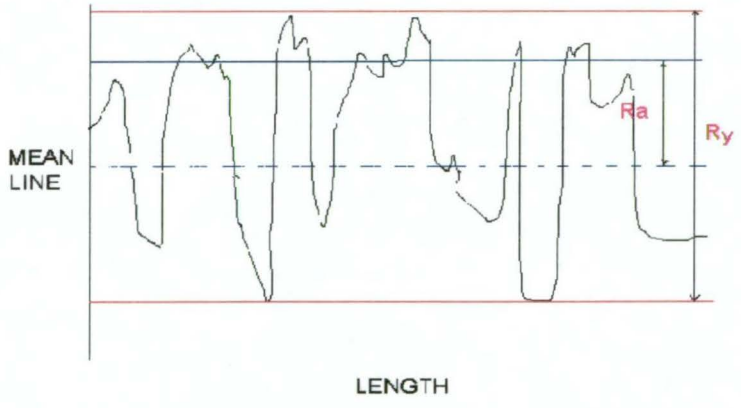

Figure 5.15 Measured surface finish components

R_y is a measure of the total depth of the surface irregularities within the sampling length and is the most direct of all surface finish values. However it is not the

complete definition of roughness, as no account is taken of the frequency of the surface irregularities.

R_a is the arithmetical average of the departure of the profile both above and below its center line over the prescribed sampling length, from figure 5.16. it may be expressed mathematically as:

$$R_a = \frac{1}{L}\int_0^L y\,dx \qquad [5.1]$$

Where:

L = sampling length

Y = vertical ordinate with respect to the centerline irrespective of sign.

The value of R_a may be determined by planometric methods by summing the areas a above and the areas b below the center line and dividing by the sample length.

$$R_a = \frac{\sum areas\ a + \sum areas\ b}{L} \qquad [5.2]$$

There is no definite relationship that exists between the maximum peak to valley height and the arithmetic mean deviation other than the true geometrical profiles. In practice, R_y/R_a may vary between 3.5 for the coarser finishes produced by turning and milling, and up to 14 for the finishes produced by honing and lapping [147].

The accuracy of the profiler is very high, with measurements taken to two decimal places of a micrometre (μm). The profiler measures over a short distance of either 0.25, 0.8, 2.5, or 8mm at a rate of 0.5 mm/s. A basic hard copy output of the surftest machine is shown in figure 5.17.

This output highlights both the quantitative surface finish as well as the qualitative surface finish in the turning operation. The qualitative surface finish as seen in figure 5.17. can be seen in the profile of the surface. The profile is taken over 2.4

mm which is stated in the cutoff and surface in this instance is qualitatively very good as the profile is seen to be repetitive. The texture of a repetitive profile is usually smoother to touch than a random profile and can usually be seen in the Ra values. The nature of the surface profile in turning operation is to take the shape of the tool, in figure 5.17 it can be seen that the tool has approached from the top of the page and has fed to the right of the page. The depth of the profile is only 17.1 at a maximum thus the profile has been left by the very tip of the tool, it can also be seen that the cutting edge of the tool is slightly worn as the profile shows a straight leading edge.

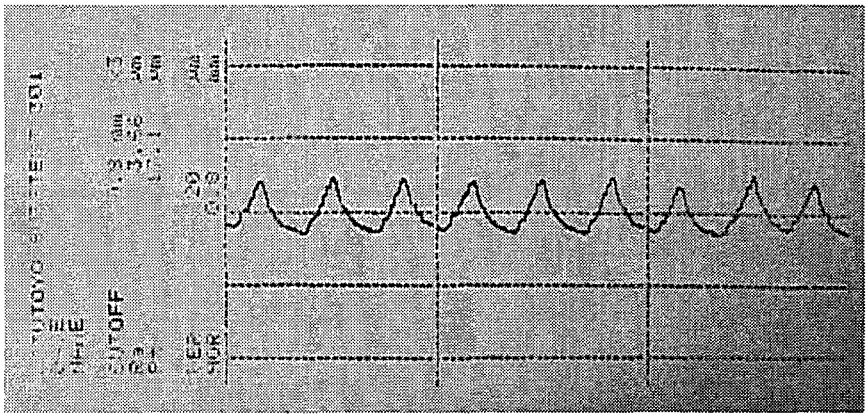

Figure 5.17 Example of a surface finish profile from Surftest 301

All of these components mentioned previously make up the first level of the experimental setup. For the completion of the setup, the system has to incorporate data acquisition, to allow for qualitative measurement. This is explained in the next section.

5.2 DATA ACQUISITION

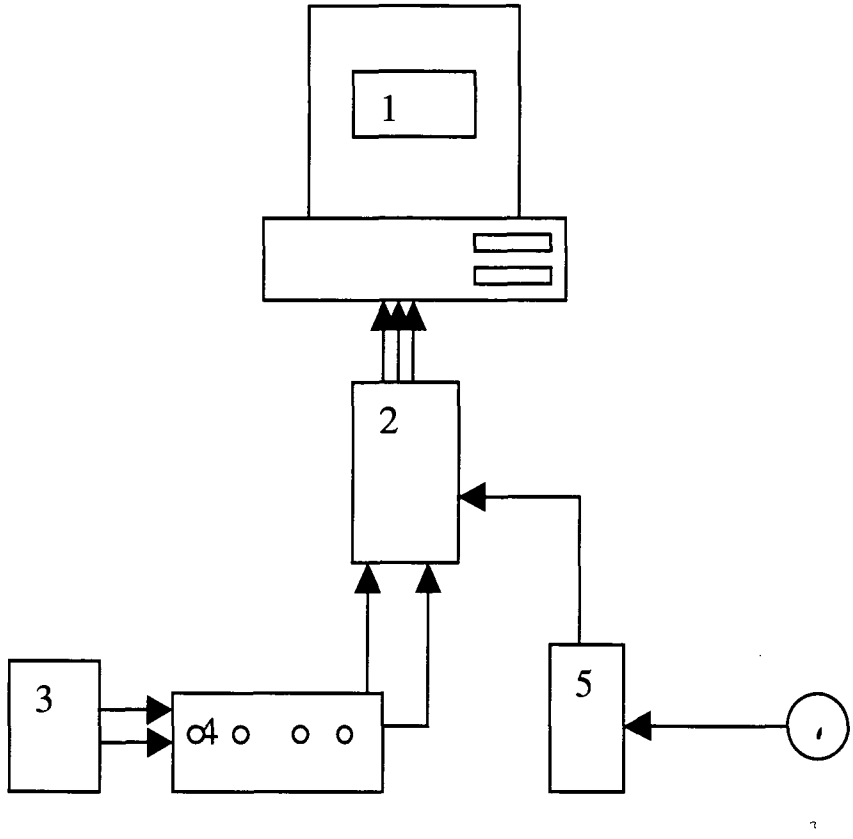

Figure 5.18 Data acquisition setup

Figure 5.18 shows all the components involved with the data acquisition setup. The components are as follows:

1. Pentium 90 processor with 32 Mb of ram. The system contains the data acquisition program called Lab View (discussed later).
2. Data acquisition card (discussed further).
3. Dynamometer.
4. Dynamometer four channel amplifier (two required as shown above).
5. Accelerometer one channel amplifier.
6. Accelerometer.

The first component of discussion is for the data acquisition card.

5.2.1 Data Acquisition Card/Board

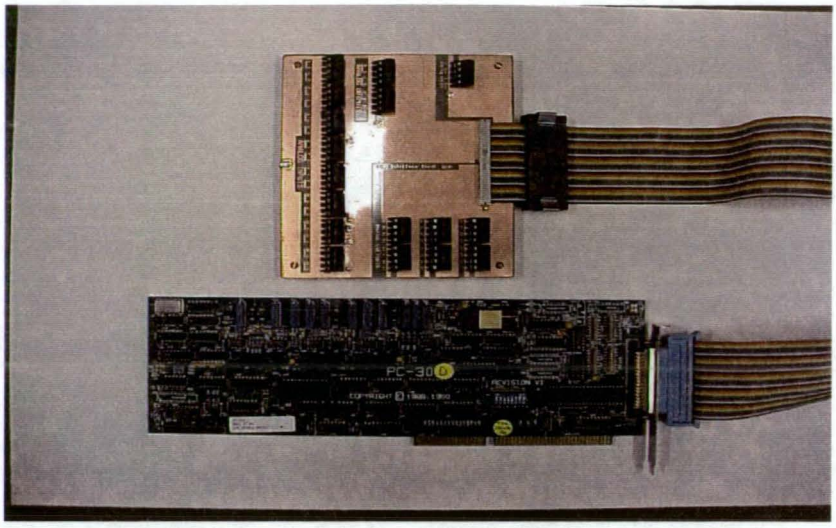

Figure 5.19 Data acquisition board and card

The data acquisition card (figure 5.19) used for the experimentation is a PC30D high performance card with a A/D throughput of up to 200KHz. It is designed for use in PC/AT and AT compatible personal computers. In addition to improved throughput, the PC30D also contains FIFO buffers for A/D data, for improved operation in conjunction with multi tasking operating systems such as windows 95. The key specifications of the PC30D card are summarised in table 5.5.

SPECIFICATION	VALUE
A/D resolutions	12 Bits
Nonlinearity	Less than +/- 0.75 LSB
A/D full scale input ranges	-5 to 5 Volts
Number of A/D inputs	16 single ended
A/D throughput rate	200KHz

Table 5.5. Data acquisition card specifications.

A/D conversions may be monitored by either polled I/O, DMA or by interrupts. Polled I/O is not reliable for use under Windows environments because it requires

full attention of the CPU (Central processing unit). For that reason, DMA (Direct Memory Access) transfer mode was used since it is the most efficient way to acquire data and can support speeds of up to 330KHz.

From figure 5.18 it can be seen that the data acquisition card requires three inputs. Two of those are the tangential and feed forces from the dynamometer amplifier, and the third is the vibration, form the accelerometer amplifier. All three of these signals are then combined in the central processing unit, and outputted through Lab View.

5.2.2 Data Acquisition Program

Labview is a programming language like C or Basic, but it differs in that it uses a graphical base as opposed to a text-based code. This graphical base is known as G code. Labview is extremely useful in data acquisition in that it has extensive libraries of Virtual Instruments (Vi's), which can be automatically used within the program code, rather than having to type in the complete code. A Vi is a section of graphical code that performs a certain task, such as taking in the data from a turning process such as in this case. While the Vi diagram has to be constructed (the overall program code), labview provides sub Vi's that can be imported into the program. These sub Vi's may be time keeping devices, voltage source readers, and other applications.

Labview consists of two main components:

1. The front panel (figure 5.20). The front panel is the user interface. This is where the data obtained from the acquisition is visualised. The front panel is created through Labview's incorporated tool and function palette. As an item is added to the front panel, its subsequent code will be added into the Vi diagram.

2. Vi diagram (figure 5.21). The Vi diagram is where the programming occurs. As can be seen in figure 5.21, the language has similar attributes to other programming languages, but rather than extensive code, it uses simple graphical components and wires to display the code. When an item is added

to the front panel, it will appear as a graphical code element in the Vi diagram. It then has to be wired up to other functions to make the front panel operate. Labview's in-built error checking system will automatically detect the source of all errors and has online help to suggest how to fix the problem.

The compiled Labview program was written in order to obtain data values for the two force components, and the frequency of vibration. The Labview front panel (figure 5.19), and the Vi diagram (figure 5.20), are illustrated below. Each diagram will be explained in detail. A working version of the program is available on the attached CD.

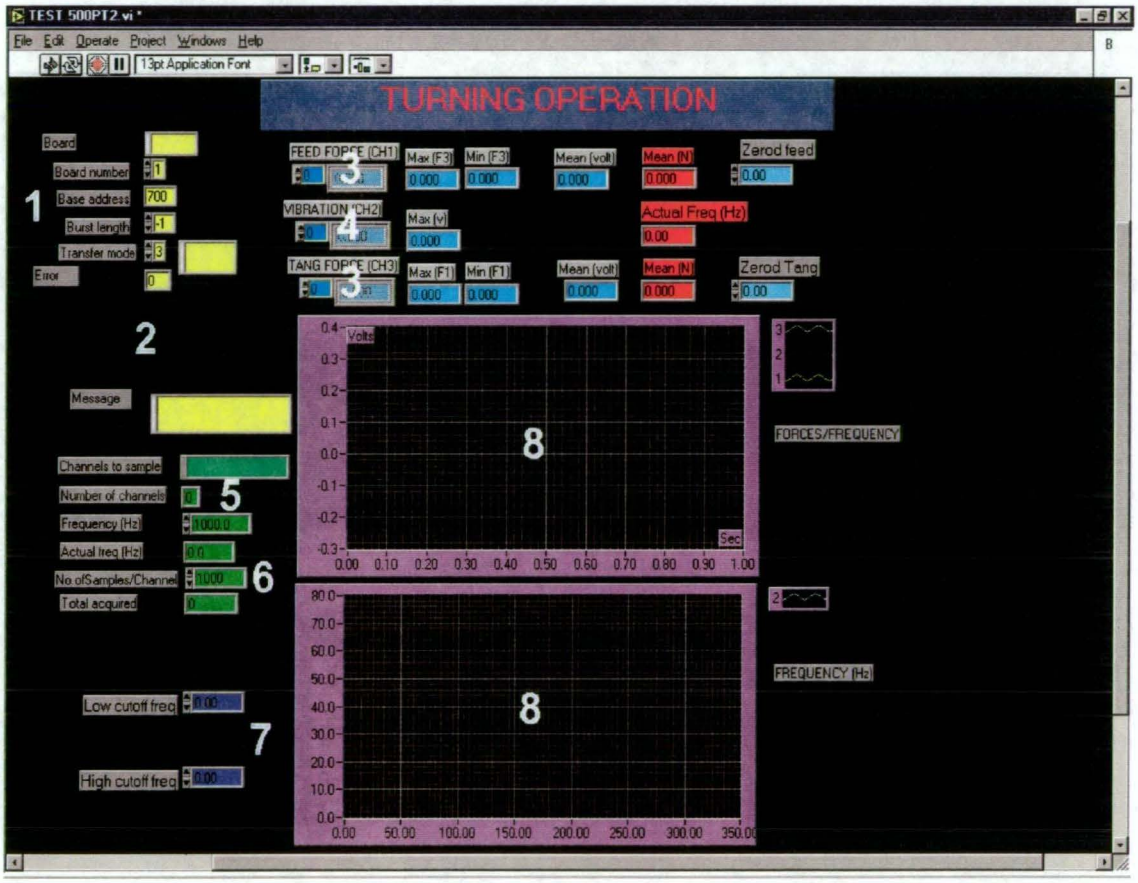

Figure 5.20 The front panel

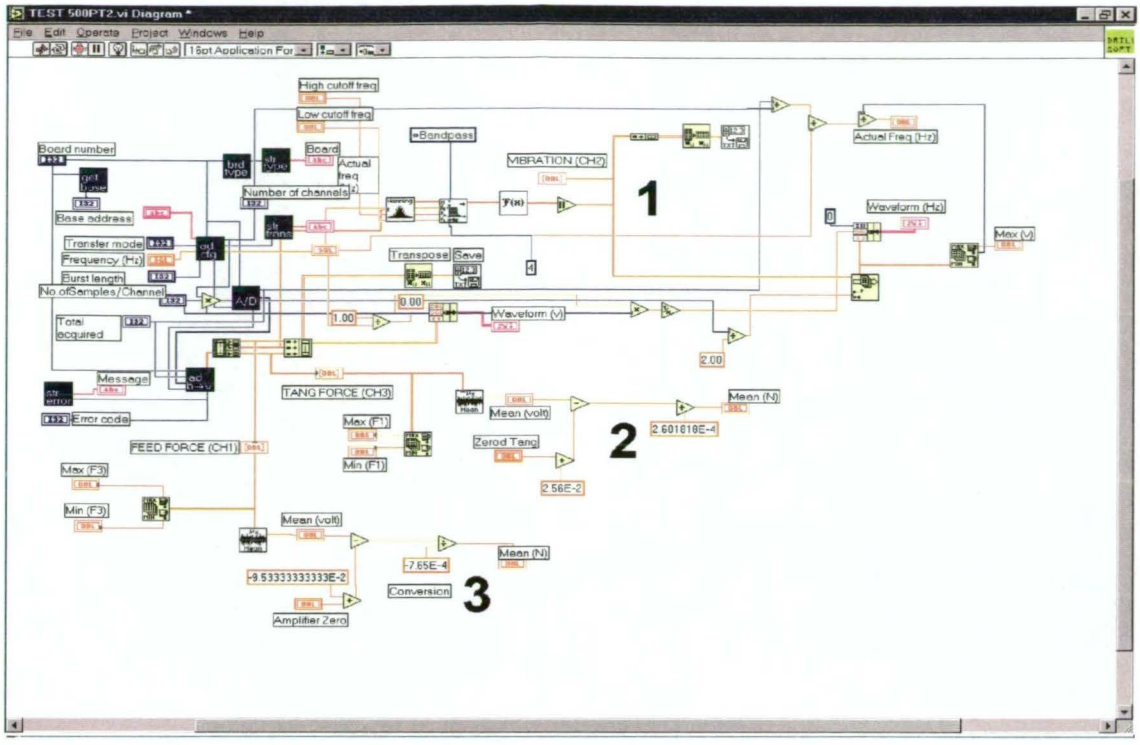

Figure 5.21 The VI diagram

The front panel (figure 5.20) gives the data output from the experimental testing. This panel has several main components. These are illustrated in table 5.5.

1. Board Details	These include board, board number, base address, burst length and transfer mode. These are automatically recognised by the system through the Getbase.vi and Brdtype.vi. This information is based on the data acquisition card present.
2. Error	If any error is present within the system, such as no acquisition board present, the system will output an error code, depending on the situation, and the Strerror.vi will execute an error message describing the problem
3. Feed Force Tangential Force	The feed force and tangential force sections have six main categories. The first is an array of the acquired values, where for 1000 samples, any individual voltage output can be viewed. The next three sections give the minimum, maximum and average voltage values recorded for the corresponding forces. The zerod values are used in conjunction with the average force in Newton's. This value is obtained from the equation of the calibrated data, and is incorporated into the labview code (explained later).
4. Vibration	The vibration section has similar characteristics to the forces. There is an array of voltages, a maximum voltage output value, and then the most important is the Actual frequency in Hertz, or cycles a second. This value is

	calculated by using a Fourier transform (in-built Labview function), to transform the infinite many voltages over a time domain, into a natural frequency. This is explained furthermore in the calibration section.
5. Channels to sample Number of channels	The number of channels to sample in this case was three. Two force channels and one frequency channel. The data acquisition board can be seen in figure 5.19. This is where the channel routes are chosen. The two force components were wired into channel 1 and 3, and the vibration wired into channel 2. From the data acquisition card, the channels are then wired into there prospective amplifiers. This information is then inputted into Labview so that the data acquisition occurs on the required channels.
6. Frequency samples/channel Total acquired	The frequency is in Hz or cycles per second. Therefore for a 1000Hz, there are 1000 samples taken per channel per second. If the number of samples per channel is 1000, then the data acquisition occurs over one second. For 10000 samples per channel, then the recording occurs over 10 seconds.
7.Low/High cutoff frequency	This information is based on a Butterworth filter. Given that irregular and excess vibrations can occur outside the range of testing, the Butterworth filter allows the application to taper out excess vibrations not associated with the testing range.
8. Display Charts	There are two charts displayed on the front panel. The first is a voltage time graph, based on the vibration voltages. The second chart displays the converted voltage signal (by the Fourier function) as a frequency spectrum. This shows the frequency versus amplitude.

Table 5.5 Main characteristics of the front panel

Table 5.5 gives a brief overview of the front panel, however the most important characteristics of the section are the three numerical boxes highlighted in red. The other details on the front panel are simply building blocks for these three sections, as is the back panel. These three values are the Actual forces in both directions (in Newton's), and the actual frequency (in Hz), which for each test run and the importance of the investigation are the required experimental data. The Labview block diagram that contains the code for the acquisition of these values will now be explained.

From figure 5.21 it can be seen that the block diagram, or source code, is broken into three main parts. The first is the code to obtain the frequency, then the second and

third are for obtaining the two force components. The process behind these three sections is explained below.

1. Frequency determination. The frequency is first obtained through a set of sub vi's (explained later) firstly as a set of voltage readings with respect to time. These voltage values are then run through Hanning and Butterworth filters to remove any unwanted or excess vibration. The signal is then processed with an in-built Fourier transform, to convert the signal to a Frequency (Hz) per sample domain. The maximum of this array is obtained to give the actual frequency in Hertz for the given situation.

2. Tangential force determination. The process is very similar to the frequency, but requires initial calibration before the actual forces can be obtained. The Labview function obtains an array of voltage readings of which an average for the test period is found. Given that there is not a large deviation from the minimum to the maximum voltage, then this average can be deemed accurate. From calibrated data (see next section), a linear relationship between voltage and Newton's is found, which is incorporated into the source code to allow the voltage signal to be converted, and displayed on the front panel in Newton's.

3. Feed force determination. This process occurs in an identical manner to the other force component. The equation of transformation to actual force differs, but this is also resolved in the calibration procedure.

It is important to discuss now the calibration procedure which occurred for each particular device, in order to make the Labview application and the experimental objective complete.

5.3 PROCEDURE ON CALIBRATION

As mentioned several times in the last chapter, the three tested variables must be reliable, otherwise the neural network model will not yield accurate results. Therefore calibration of all the devices must be performed.

5.3.1 Steps Taken for Surface Profiler

The calibration of the surface tester is a simple procedure. The calibration of the surface profiler is carried out prior to testing, and is done so by using a supplied specimen seen in figure 5.14. The specimen has a given surface roughness of Ra = 2.9 um +/- 3%. The gain of the tester is adjusted until the displayed value equals the given roughness value.
While this is difficult to achieve precision in, the analyser was calibrated to 2.89 um.

5.3.2 Steps Taken for Dynamometer

The dynamometer is a lot more difficult to calibrate. The process was achieved via a Shimadzu tensile testing machine. A point load was then applied to a dummy tool, at 40mm from the front of the dynamometer (the same distance as where the tool will be). The dynamometer was then placed in the loading machine, and in conjunction with the Pentium 90 processor and the labview software, the forces were recorded.

Two basic calibration tests had to occur, because of the need for calibrated data for the feed, and the tangential force components. To test the two force components, the dynamometers orientation with respect to the force were simply changed. Figure 5.22 shows the calibration testing for the tangential force component.

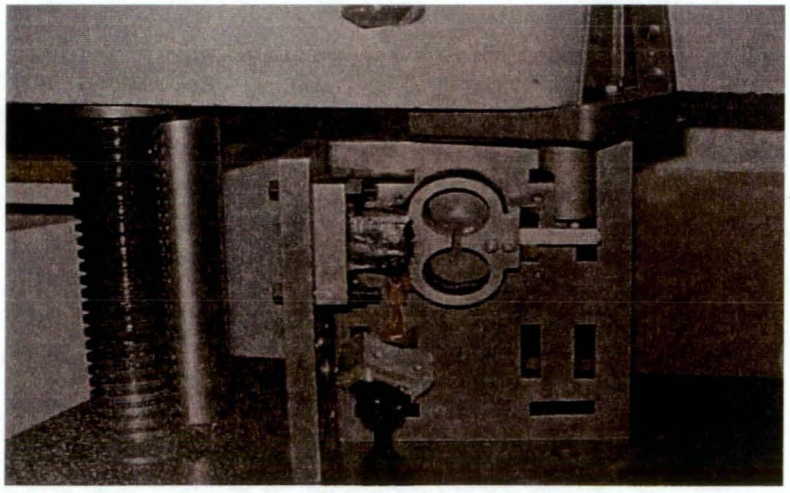

Figure 5.22 Dynamometer calibration

The tests were done over a range 0 N to 1200 N, firstly in 10N increments, up until 200N, then in increments of 200 N. The reason for the need for so many testing data points, is that labview outputs the force readings as a voltage, not as a Newton value. Therefore a linear relationship needs to be found between voltage and Newton values. The calibration test data can be seen in the graphs in figure 5.23 and figure 5.24.

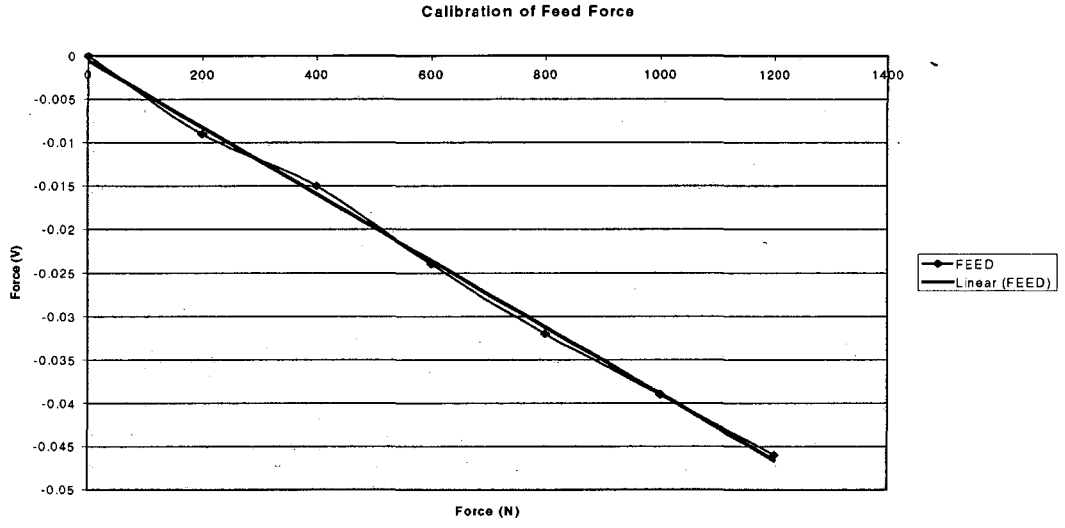

Figure 5.23 Calibration data for feed force

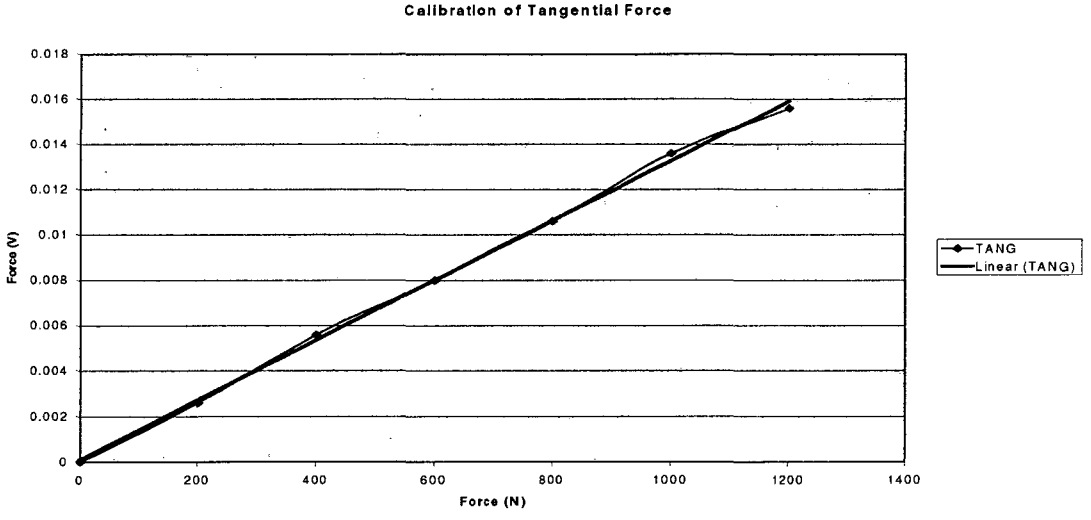

Figure 5.24 Calibration data for tangential force

The linear relationship between the voltages and force components is:

TANG FORCE (V) = 2.827E-04* TANG FORCE (N)

FEED FORCE (V) = -8.122E-04 * FEED FORCE (N)

These equations may seem to have to too many accurate digits, yet the forces in Newton's are to the order of E+3 and greater, hence you are multiplying powers of E-04 by E+03. Therefore even the smallest approximation or rounding off of the actual relationship will cause large inaccuracies.

Now that the relationships have been found between voltages and Newton's, the Labview code can be altered so that the front panel actually outputs the force in Newton values.

5.3.3 Steps Taken for Accelerometer

It was previously mentioned that the Labview application, outputs frequency values in voltage. These voltage values must be converted to natural frequency (Hz). This can be done internally in Labview by using a Fourier transform function (explained in 5.2.2). Even though the labview application is set up to out put vibration in Hz immediately to test if the values are correct, a vibration of known frequency is required. This is done, by using a vibrating beam of known frequency. Given that a 50Hz (for example) excitation is initiated, then the Labview application should be able to output the same frequency on its front panel. This process is repeated until the data acquisition conforms to all known frequencies.

The experimental setup for the calibration of frequency and conformance of the Labview application is shown in figure 5.25.

Figure 5.25 Calibration of frequency and associated software

5.4 COMPREHENSIVE EXPERIMENTAL TEST RANGE

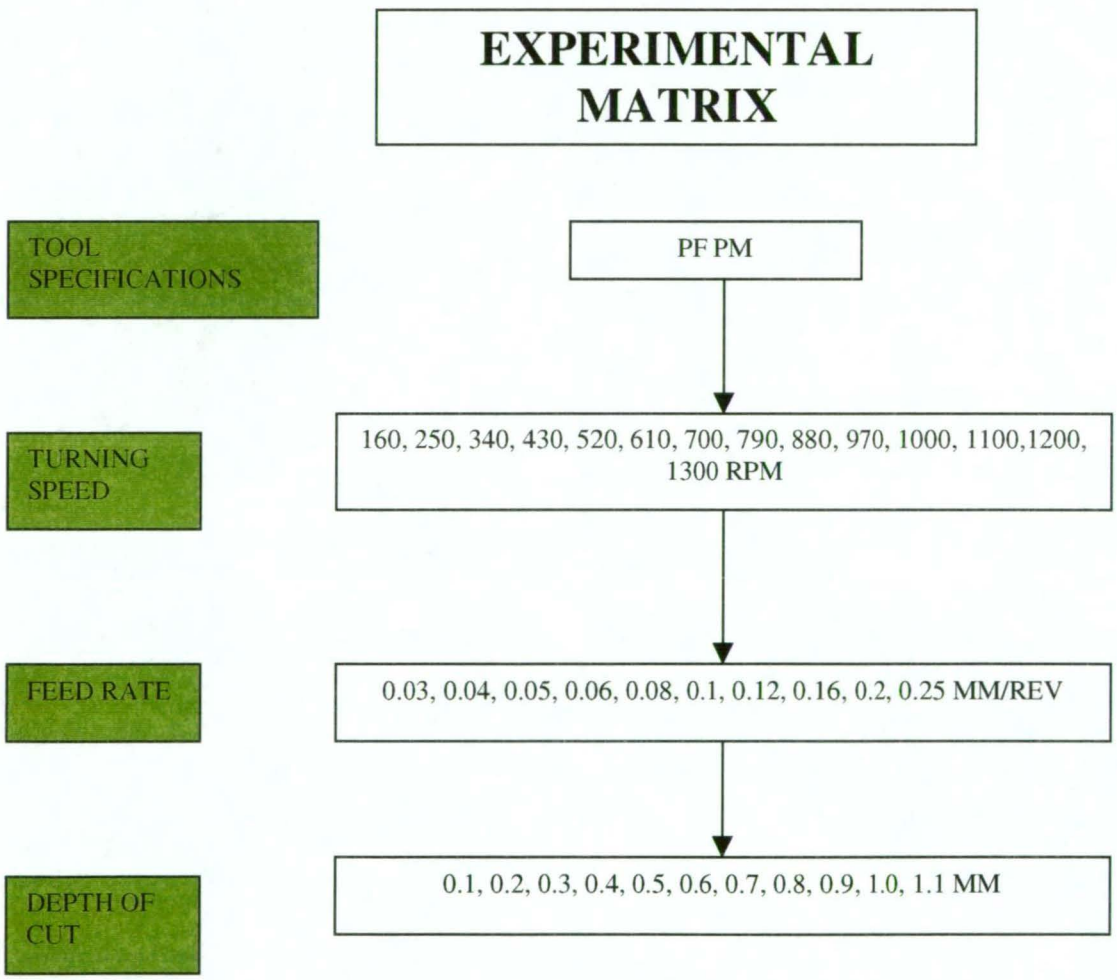

Figure 5.26 Variable ranges in testing conditions

The testing matrix, as illustrated above (full model as attachment C), is the indication of how test cuts are to be taken. The test matrix is imperative, as a true indication of the amount, size and conformity of data must be known before testing even begins. The matrix assures that the desired variables ranges can be achieved by the specified lathe, and there are no clashes with maximum values that could cause immediate tool wear and unplanned surface finish irregularities.

The matrix starts with the tool types. Each tool type will be used over the entire range of variable tests. Three of each tool type are available, so generally the tests will be broken into three major section. Each major section will incorporate three minor subsections. This will incorporate the alternation of the feed speed and depth of cut variables. While one variable is changing, the other variables will remain constant. The variable constant values are:

1. Feed 0.08 mm/rev

2. Depth of cut 0.5 mm

3. Speed 300 rpm

For example if the feed and depth of cut are being held constant at the above values, then the feed will begin at 160 rpm for the first cut section, and then increase after each section, up until its maximum of 1300 rpm. The process will then continue with one of the other variables alternating while the remaining two become the constants. The output from this testing matrix is represented in table 5.6.

TEST	TOOL	FEED (mm/r)	SPEED (rpm)	DEPTH OF CUT (mm)	Ft (N)	Ff (N)	f (Hz)	Ra (µmm)	Ry (µmm)
1	PF	0.03	300	0.5					
2	PF	0.04	300	0.5					
3	PF	0.05	300	0.5					
4	PF	0.06	300	0.5					
5	PF	0.08	300	0.5					

6	PF	0.1	300	0.5					
7	PF	0.12	300	0.5					
8	PF	0.16	300	0.5					
9	PF	0.2	300	0.5					
10	PF	0.25	300	0.5					
11	PF	0.08	300	0.1					
12	PF	0.08	300	0.2					
13	PF	0.08	300	0.3					
14	PF	0.08	300	0.4					
15	PF	0.08	300	0.5					
16	PF	0.08	300	0.6					
17	PF	0.08	300	0.7					
18	PF	0.08	300	0.8					
19	PF	0.08	300	0.9					

Table 5.6 Experimental matrix

The red data shows the first test section, where the feed rate of the lathe is altered while the other variables are kept constant. Similarly the blue text shows the variation of the depth of cut. The green region is where the first test output data is obtained. This is for the feed and tangential forces. The yellow output region is for the frequency of vibration of the tool post. Both the green and yellow regions of data will be obtained through labview (previously mentioned). The final region is the violet region, and of most importance. These are the two surface finish values, which are initially obtained via a surface finish analyser, offline. The three output regions, green, yellow, and violet, will all become inputs for the neural network to eventually obtain an online prediction to the surface finish value.

This extensive procedure for the implementation of experimentation, paves the way for quality conforming results. Since the equipment was calibrated with such a high degree of accuracy, the error output expected from the test variables and results is expected to be of a minimum nature. The results of this experimentation are now summarised in the next section.

CHAPTER 6

QUALIATIVE AND QUANTITAVE TRENDS

The previous chapter explained the extensive test setup and calibration. After testing was completed the test data was tabulated and can be seen in Appendix C and D. Appendix C shows the complete raw test data with both voltage readings recorded and Labviews output data, as well as surface test results. Appendix D displays the data required for qualitative and quantitative analysis.

Over 200 test cuts were taken during the experimentation. Each test cut occurs over an average length sample of 20cm with a diameter of 8cm. These test cuts focused on the process variables of:

1. Tangential Force
2. Feed force
3. Arithmetic mean surface finish
4. Maximum peak to valley height surface finish
5. Frequency of vibration

These test parameters now allow for an analysis of the results, and also gives a chance to complement existing trends in the literature and improve the confidence in the equipment used. These effects of the process variables on the performance are dealt with separately as qualitative and quantitative effects.

The effect of feed rate, depth of cut and surface speed on forces, surface finish and vibrations is tested to verify the qualitative trends. The quantitative trends highlight the effect of the hardness and variety of cutting tool, in both the conformance characteristics and the separating factors. Both sections will show the reliability of the testing equipment, and the repeatability of the results, giving reference to present trends and also revolutionary new trends.

6.1 QUALITATIVE TRENDS

6.1.1 Effect of Feed Rate

The first qualitative step is to analyse the effect that the feed rate has on the five experimental outputs. The following graphs display the feed rate with respect to the other recorded variables. The graphs are taken from data displayed in appendix D.

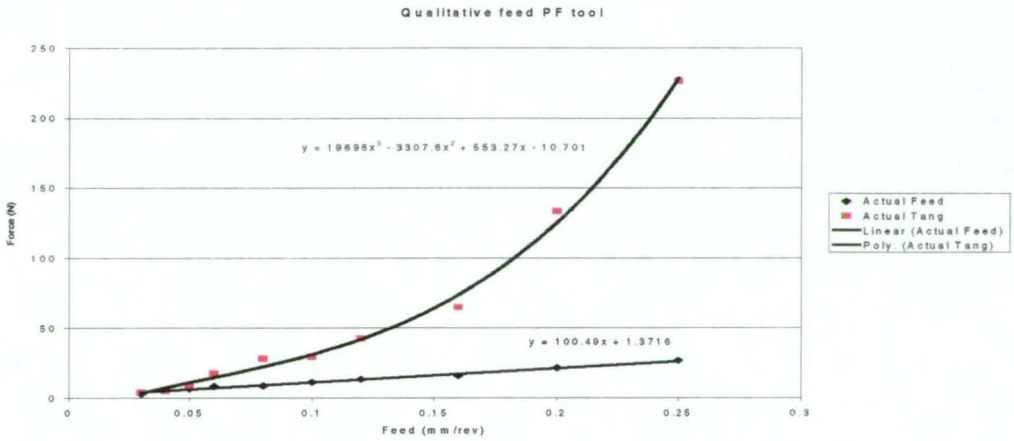

Figure 6.1 Effect of feed rate on forces

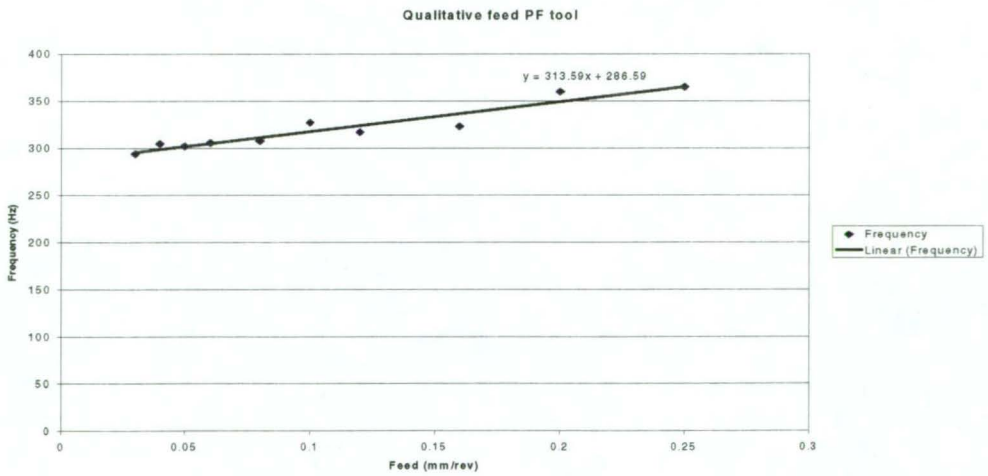

Figure 6.2 Effect of feed rate on frequency

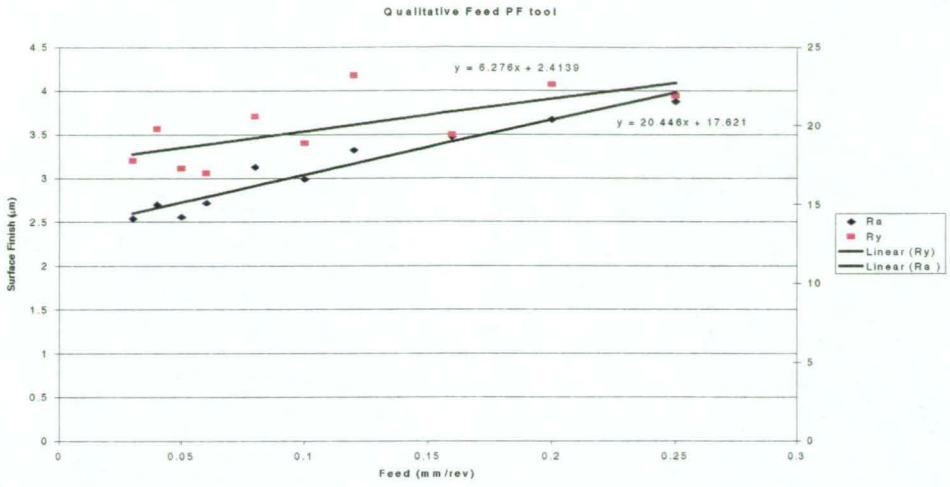

Figure 6.3 Effect of feed rate on surface finish

It is known that the feed of operation is directly related to the cutting forces. Literature suggests [3,6] that as the feed rate increases the forces increase proportionally. Likewise figure 6.4 from Production Engineering [148], shows the trend of the three force components, Tangential (Pz), Radial (Py) and Feed (Px), with respect to cutting feed.

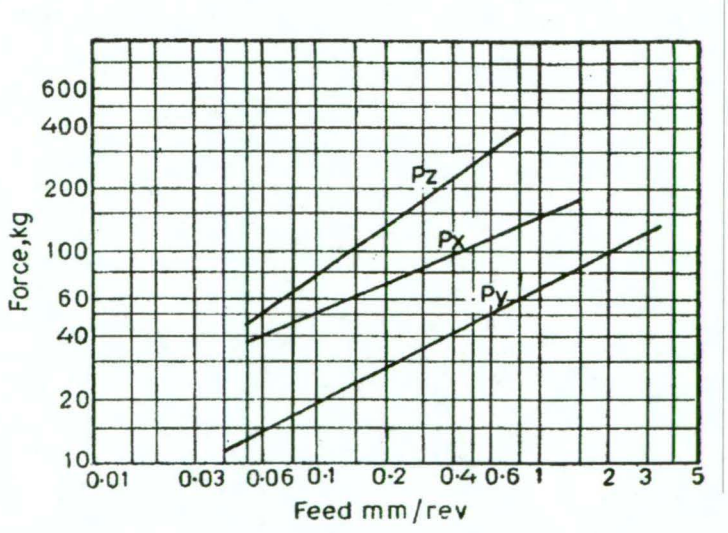

Figure 6.4 Force trends from literature [148]

From the experimental data (figure 6.1), the trend can be seen to follow those prescribed in literature [148]. As the feed rate increases, so to does the Tangential

and feed forces. It can be noted also that the tangential forces increases at a greater rate than the feed force with increasing feed.

As expected, with increasing forces the frequency of vibration also increases with increasing speed. Given that the forces and frequency are increasing, the expected result would be the decay of surface finish quality. Figure 6.3 displays this characteristic, with the Ra surface finish values decaying at a much greater rate than those of Ry. This implies that the deviations in the surface finish profile become more irregular as the feed is increased.

This gives us the overall Qualitative picture, that for an increasing feed, the forces, vibrations and surface finish also increase, and hence an increase in power consumption. This is particularly evident at high feed rates where the tangential force increases at a far greater rate.

6.1.2 Effect of Surface Speed

Literature suggests that the speed of revolution of the spindle generates little effect on the output variables. The following figures demonstrate the trends of machining when the surface speed is increased while the other variables are kept constant.

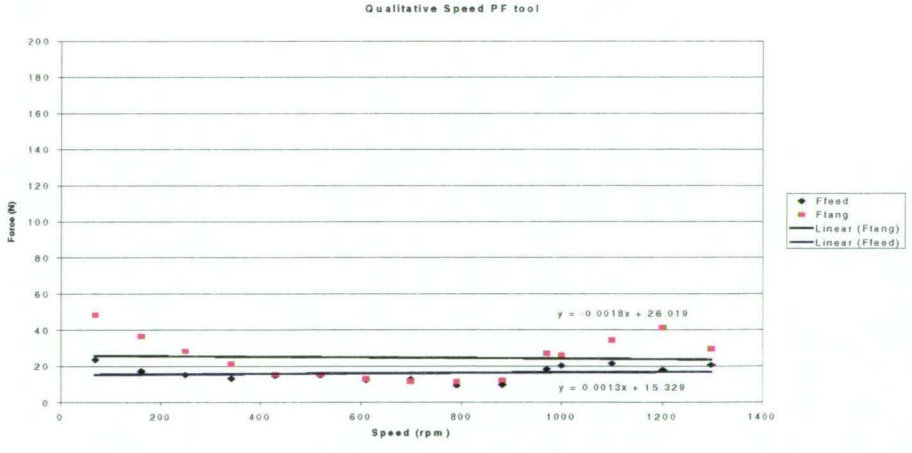

Figure 6.5 Effect of speed on forces

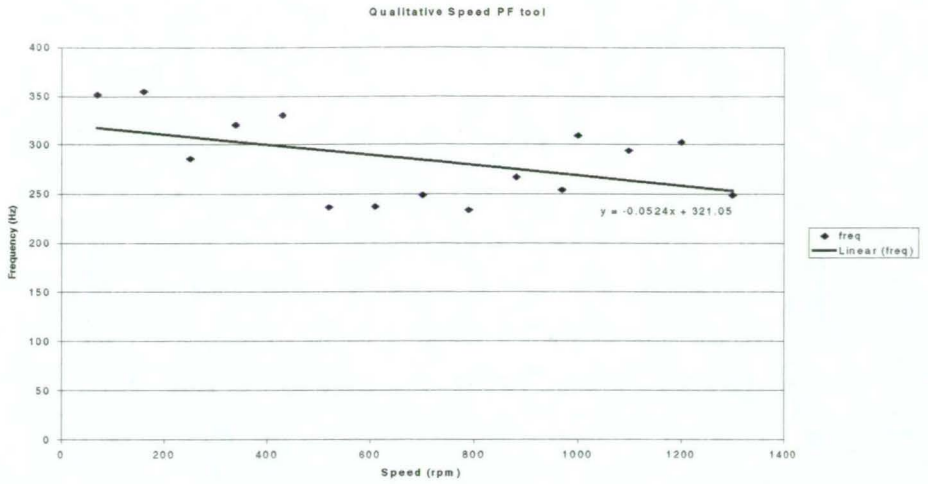

Figure 6.6 Effect of speed on vibrations

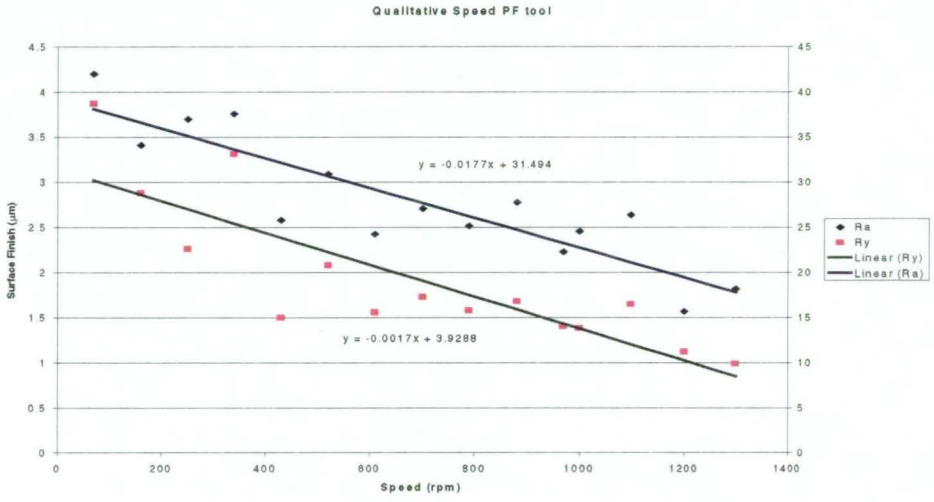

Figure 6.7 Effect of speed on surface finish

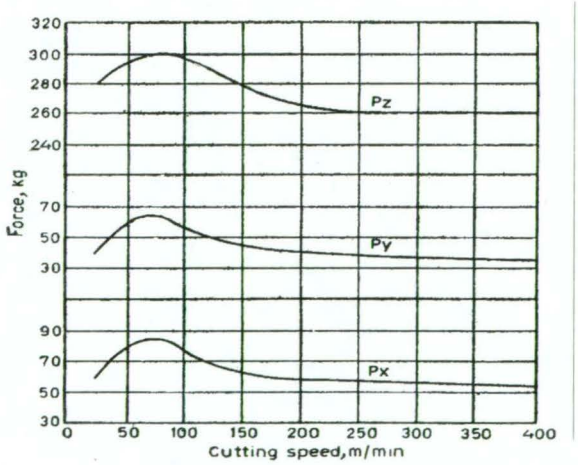

Figure 6.8 Literature trends of varying speeds [148]

It is apparent that for low speed operating condition, slight changes in the forces are apparent, but at higher speed, the forces become independent of speed changes (figure 6.8). This trend will certainly depend on the composition of the work material. S1214 steel (similar to the steel used in experimentation) and 65s-T6 aluminum will be literally independent of force variances over all speed ranges. Given the data trends in figure 6.5, it is very apparent the mild steel used in testing behaves very well in terms of material properties, and is shown to be independent of speed changes.

While it is not a dramatic change, the frequency of vibration is shown to slowly decrease over the range of increasing speeds (figure 6.6). This displays that as the speed increases, the machine is approaching a steady state in terms of machine tool vibration.

Figure 6.7 shows that the surface finish is very much effected by the alteration of cutting speed. As the speed is increased, the frequency reduces, while the forces remain constant, therefore causing the quality of the surface finish to increase. Therefore a products quality and conformance will increase as the speed rate is increased (over these specified speed ranges.

Given that:

Power = Force * Velocity [6.1]

The power in the system is still increasing with increasing speed, yet at higher power consumption rates the system is quite obviously approaching steady state machining.

6.1.3 Effect of Depth of Cut

In studying the effect of the depth of cut with respect to the other variables it has been found that increasing depth of cut will increase the other operating conditions of the system. The following charts demonstrate the trends of machining when the depth of cut is increased while the other variables are kept constant.

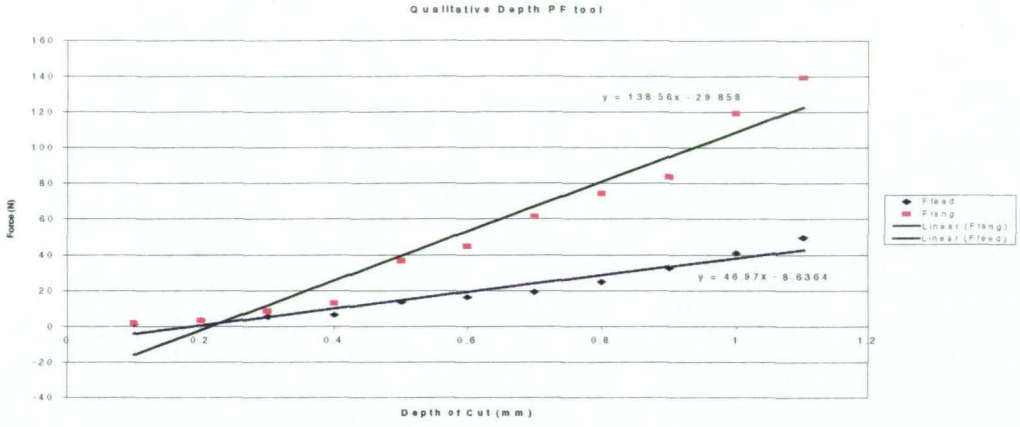

Figure 6.9 Effect of depth of cut on forces

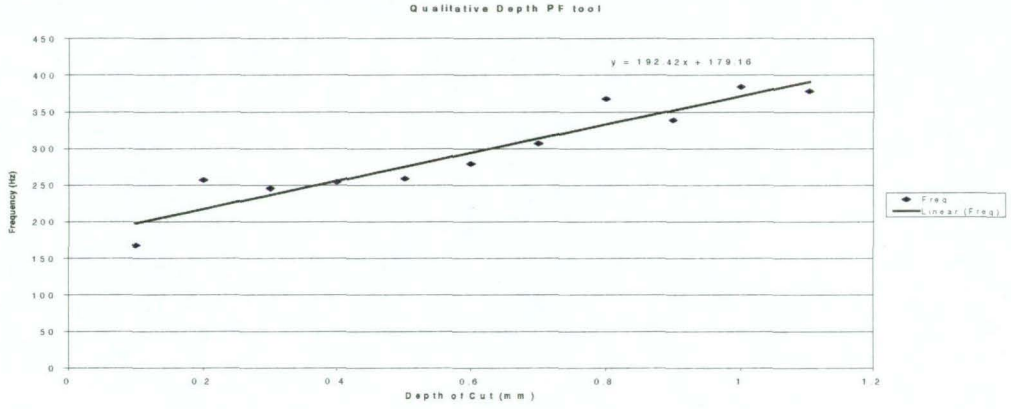

Figure 6.10 Effect of depth of cut on vibrations

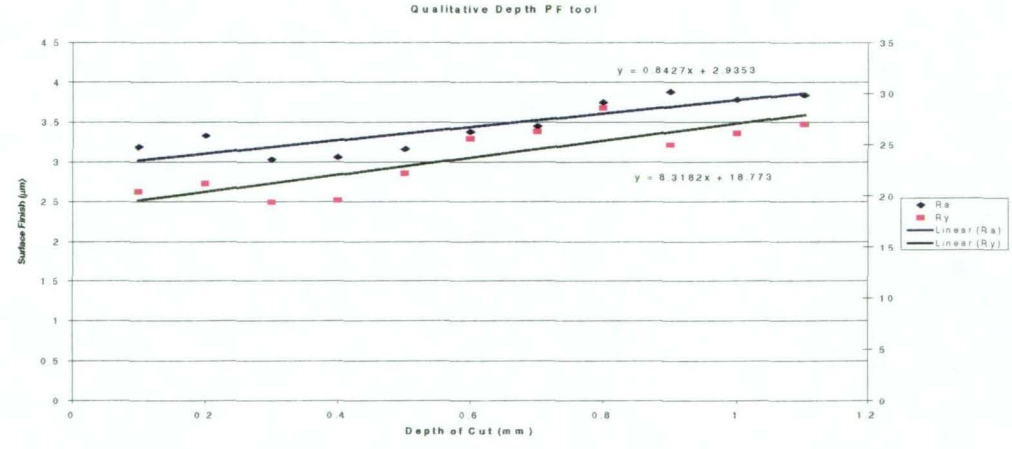

Figure 6.11 Effect of depth of cut on surface finish

Literature [148] conveys that depth of cut does not have much effect on the specific cutting resistance. But the tangential force component increases in the same proportion as the depth of cut, i.e. if the depth of cut is doubled, the tangential force also gets doubled. For a 90 degree approach angle, the feed force increases almost linearly with depth of cut, but there is practically no change in the value of the radial force.

The test data displayed in figures 6.9- 6.11 conform precisely with this literature. From figure 6.9, at a depth of cut of 0.6 mm, the tangential force is around 60N. Given a doubled depth of cut of 1.2 mm, we can see the tangential force trend displaying around 120N. This is precisely as the literature suggests.

Given that there is a linear increase in the force components, the graphs display the expected results, in that as the forces increase, so to does the frequency (figure 6.10), and likewise there is a decay in the quality of surface finish values (figure 6.11).

From equation 6.1, it can be stated that while the speed and feed rates remain constant, the forces are increasing. Therefore the power utilised by the system is also increasing, but not to the level achieved by large feed rates.

The qualitative diagrams shown above are all for the test data associated with a PR, or finishing tool. The qualitative analysis associated with the medium tool can be seen in appendix 6. These diagrams exhibit the same characteristics, which will be evident in Quantitative analysis.

6.2 QUANTITATIVE TRENDS

The previous section examined how the four input process variables effected the five experimental outputs. This section deals with how these factors effect the reliability of system when varying the cutting tool types. It gives a quantitative look at varying the machining operation. For this experimental procedure two varieties of tools were investigated. Namely a finishing tool (PF) and a Medium tool (PM). The details of these tools can be seen in section 5.1.2.

6.2.1 Effect of Feed Rate

Quantitatively it would be expected that the feed rate will generate similar trends between the tool types, but with a noticeable separation. The changing feed rate is the most influential parameter on the process (predictive importance). This is discussed in detail in chapter 7.6. The following figures display the quantitative trends for the two tool types while varying the feed rate of the system. This is done while keeping the speed and depth of cut constant.

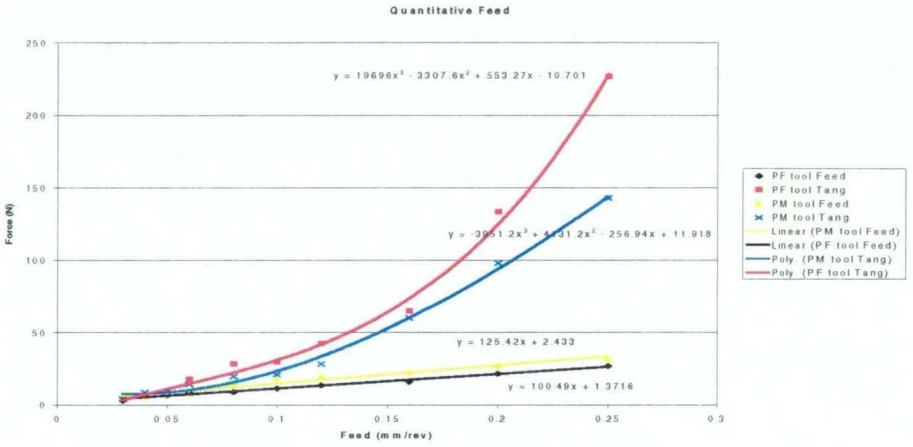

Figure 6.12 Effect of feed rate on forces

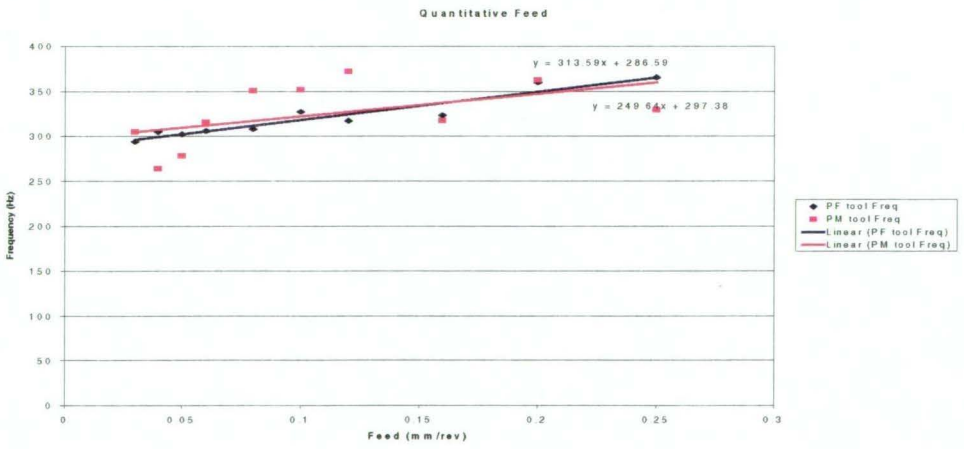

Figure 6.13 Effect of feed rate on frequency

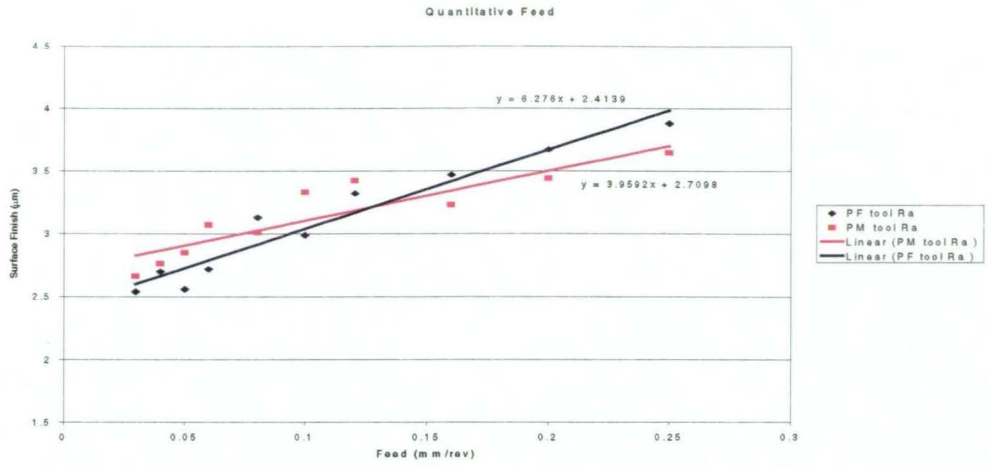

Figure 6.14 Effect of feed rate on surface finish (Ra)

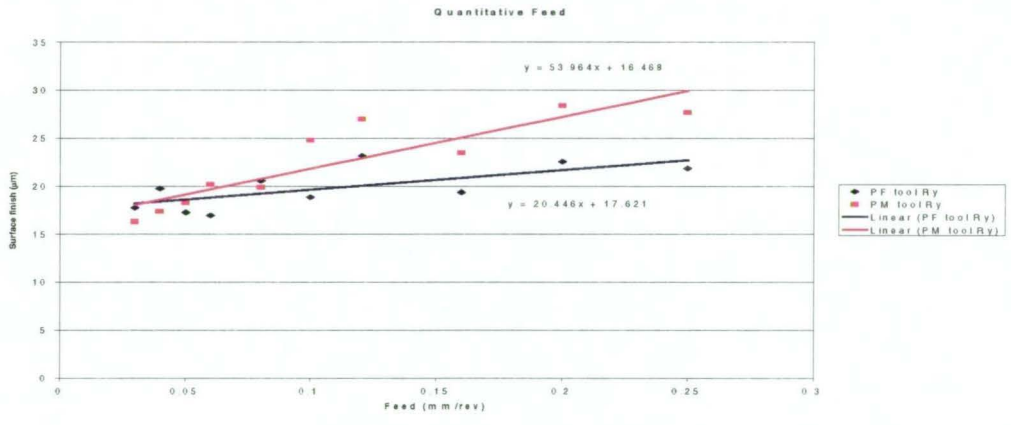

Figure 6.15 Effect of feed rate on surface finish (Ry)

The trends displayed in the above diagrams show the excellent conformance of both the cutting tools and the work piece material. While differences are expected to set the tool types apart, the general trends should exhibit the same characteristics.

Figure 6.12 displays the variance in force components with changing feed for both tool types. While there is very little difference in feed force, the tangential force shows a significant trend. While the forces of both tools increase almost exponentially, in the same fashion, it can be seen that the softer tool (PF) experiences higher forces at higher feed rates. For lower feed rates the tools generally behave in the same manner. Given that the forces exhibit similar patterns of increase, it would be expected that the vibrations of each system follow suit. Figure 6.13 shows the

conformity between the two tools. The trend lines are almost identical, but for the small difference, that the PM tool increases at a slightly smaller rate. This result conforms to those already established in literature.

Figure 6.14 and figure 6.15 are the most important to this field of study. They display the surface finish quality generated by each tool type over a variety of feeds. As expected, the finishing tool creates a more quality surface finish for low feeds, however at a feed rate of around 0.14 mm/revolution the finishing tool performs with less reliability than the medium tool. By viewing figure 6.12, it can be seen that the forces reflect this trend. At this feed rate the forces become dramatically higher for the finishing tool. Also figure 6.13 shows that at this particular feed rate the frequency of vibration for the finishing tool becomes higher than that for the medium tool.

This data gives an excellent indication of the performance characteristics of the two tool types. It also gives an insight into new approaches for industry. While it is obvious that a finishing tool is required for applications, it can be clearly seen that continuing to utilise this tool beyond feeds of around 0.14 mm/revolution will cause a rapid reduction in machining quality and will marginally increase the power consumption of the process.

6.2.2 Effect of Speed

It was shown qualitatively, that the speed rate has a minimal effect on the cutting conditions. This trend is therefore going to be mirrored in the quantitative trends.
The following charts display the quantitative trends for varying speed while keeping the depth of cut and feed rate constant.

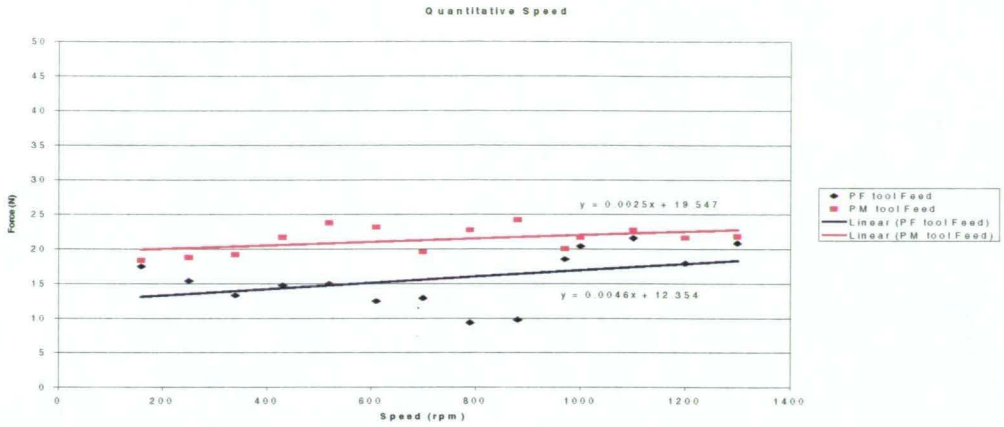

Figure 6.16 Effect of speed on feed force

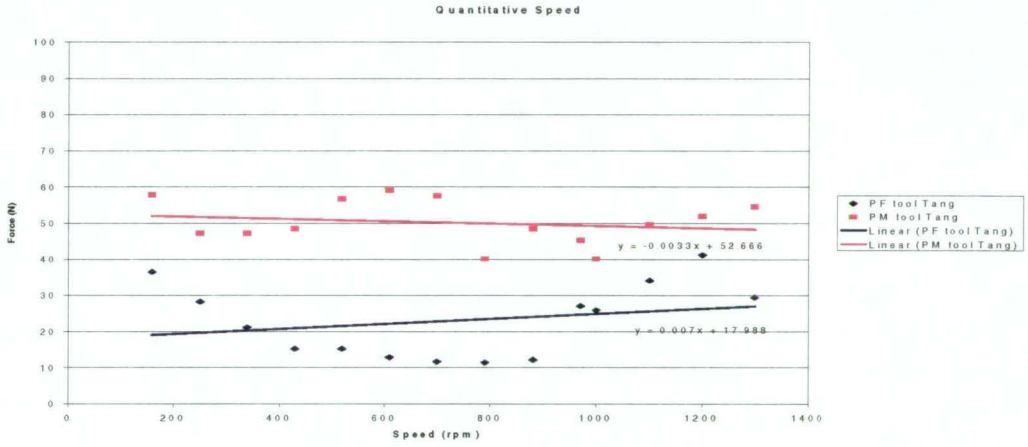

Figure 6.17 Effect of speed on tangential force

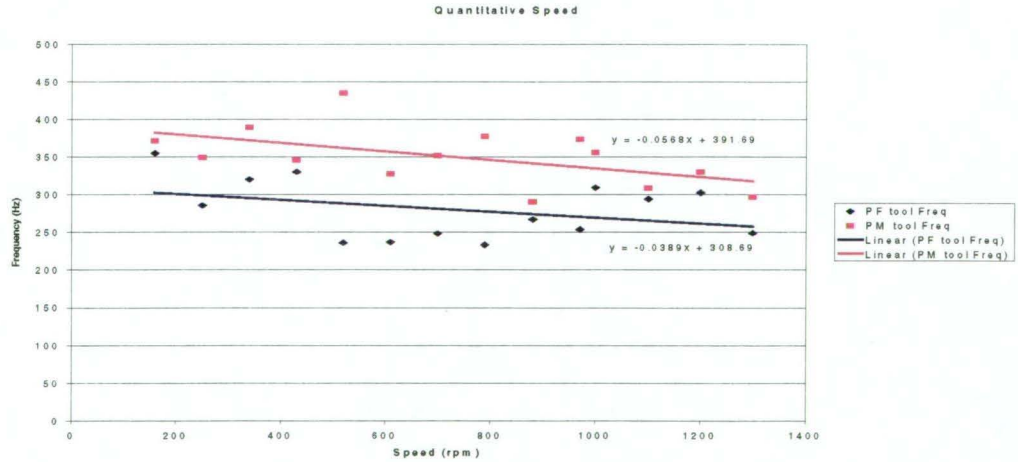

Figure 6.18 Effect of speed on frequency

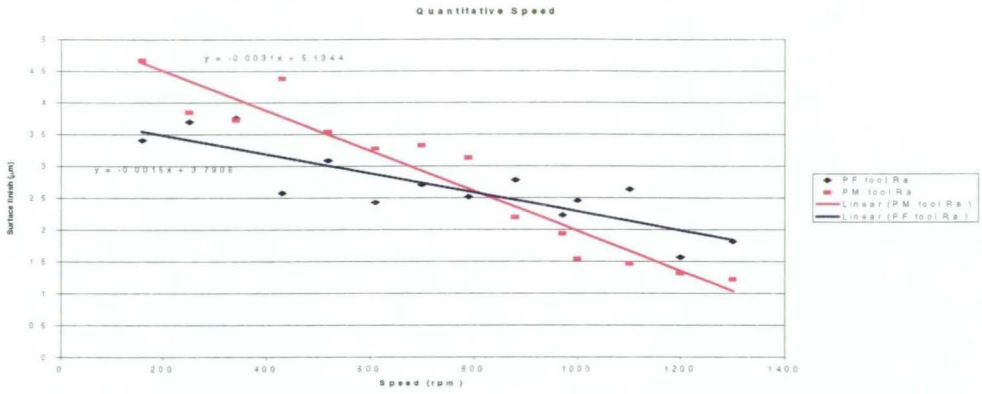

Figure 6.19 Effect of speed on surface finish (Ra)

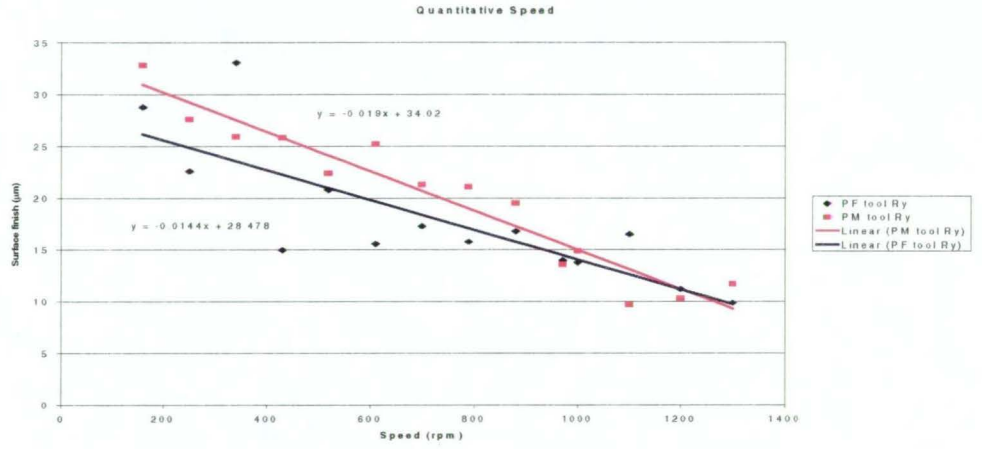

Figure 6.20 Effect of speed on surface finish (Ry)

Figure 6.16 and figure 6.17 show the two force components of the two tools respectively. Both prove that the forces of cutting are generally unaffected by changing the cutting speed, and that the softer tool (PF) operates at a constantly higher level of force than the harder tool. This is also reflected over into the frequency (figure 6.18), where the softer tool vibrates at a constantly parallel rate. The two frequency values slowly decrease over increasing speeds, reflecting the quality of instrumentation and the behavior of the test material.

Once again the main concern of interest is in the surface finish characteristics (figure 6.19 and figure 6.20). In a good contrast to varying feeds, the varying speeds follow a very similar trend. The surface quality is much better with the finer tool over the range of 0 to 800 rpm, but then at this point and onwards the PM tool generates a more superior surface finish. Figure 6.20 shows that the mean peak to valley height

relationships between the two tools. While there is a small difference in the trend lines, Quantitatively the surface irregularities decrease in size with respect to speed, at a similar rate. Once again this develops a very reliable model for industry. Given that power consumption is a key element, then utilising a PM tool at higher speeds to achieve a better surface finish, will ensure a much more efficient and cost effective manufacturing process. Given also that the tool life of a softer tool is exponentially less than that of a harder tool, the minimum cutting time required for a PM tool can be estimated through the above analysis.

This quantitative analysis contributes to previous literature, and confirms the minimal effects the speed rate has on the cutting conditions.

6.2.3 Effect of Depth of Cut

Qualitatively the depth of cut has a large impact on all cutting conditions and hence, in quantitative terms the depth of cut will present a clear distinction between the tool types.

The following charts represent Quantitative approach to cutting, with varying cut depths, while having speed and feed variables fixed.

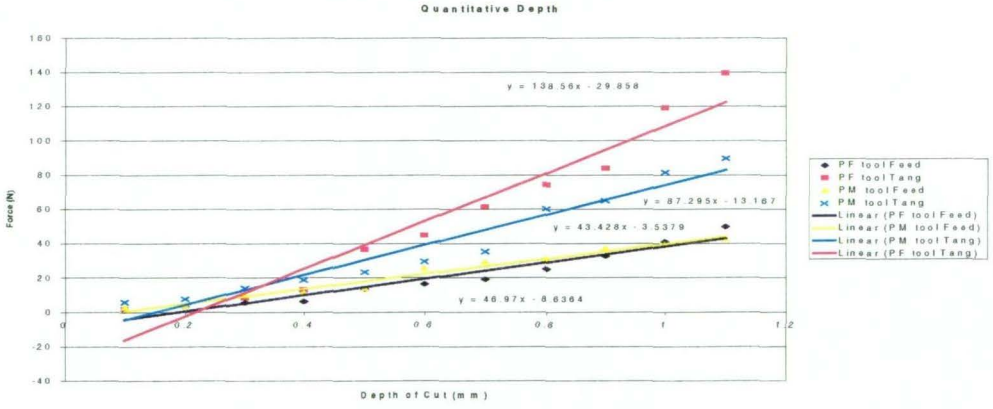

Figure 6.21 Effect of depth of cut on forces

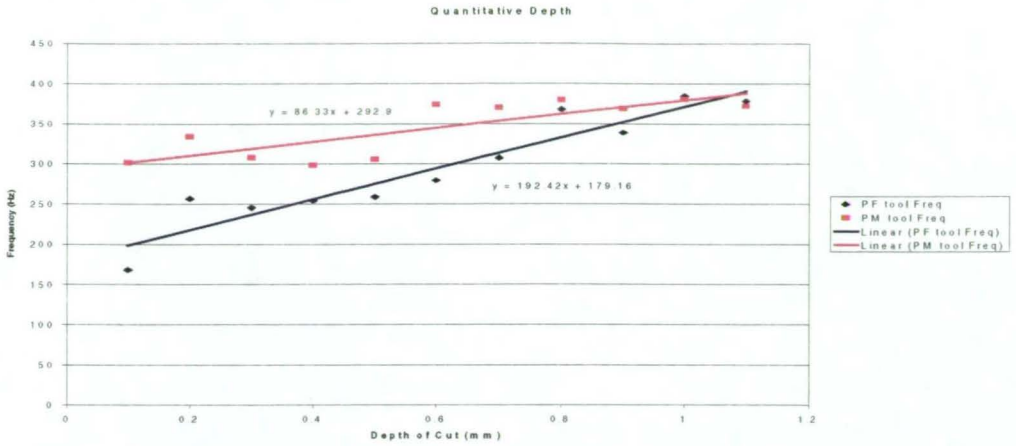

Figure 6.22 Effect of depth of cut on vibrations

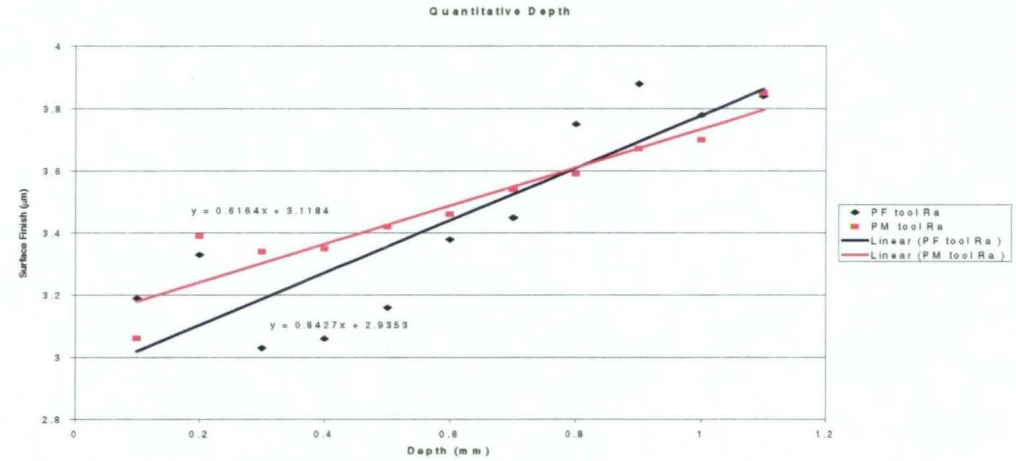

Figure 6.23 Effect of depth of cut on surface finish (Ra)

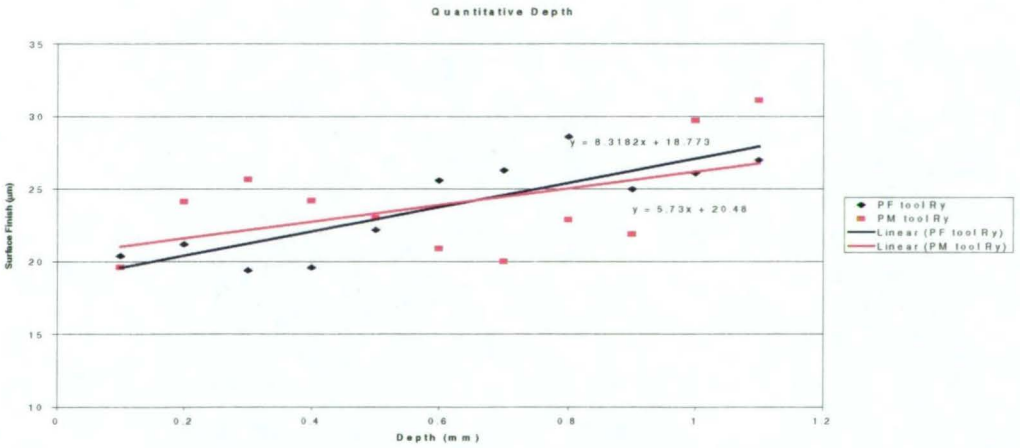

Figure 6.24 Effect of depth of cut on surface finish (Ry)

The final element of discussion is the effect varying depth of cut has on the process parameters. In actual fact, the findings suggest that changing the feed rates cause very similar variable trends to altering the depth of cut. Figure 6.21 shows that for small depth removals the forces for the PM and PF tools are very similar, but over larger increments the finer tool causes much larger forces than that of the medium tool. The feed forces for both tools vary, but not enough to suggest a concise difference between them.

Figure 6.22 shows once again, that given the forces for the finer tool are higher, the frequency is also higher. This occurs up until very high depths of cut, where the frequency of vibration for both tools becomes equal.

Figure 6.23 and figure 6.24 cement the idea that the finer tool has a limited range for which it can achieve a higher quality surface finish. At a cut depth of about 0.7 mm, the finer tool no longer has an advantage over the medium tool, and the medium tool begins to achieve a finer performance. Once again it can be deduced that, depending on the surface quality required, the finer tool only has a restricted range for which it is most efficient. Given that the power consumption for that tool is higher than for the medium tool, it may be more beneficial from an industry point of view, to negotiate the use of the finer tool.

Table 6.2 summarises the critical points for which the surface quality of the material is achieved through the alternate tool.

Variable	Value	Tool change
Feed	0.14 mm/rev	PR-PM
Speed	800 rpm	PR-PM
Depth of cut	0.7 mm	PR-PM

Table 6.2 Critical points for surface finish

This information is of a substantial benefit to industry users who are machining with softer tools over various ranges, thinking they are receiving the most desirable

performance. Probably the two most important aspects in manufacturing, are the cost of the product and the quality of the product. This project relates directly to both these aspects, in that it sets to optimise the surface quality and to reduce the power of the system by incorporating the most efficient tool type. Table 6.2 is a reference to achieving both of these criteria.

6.3 CONCLUDING REMARKS

The main focus of the experimentation was to obtain a qualitative and quantitative relation between the experimental variables, feed, speed and depth of cut, with relation to the experimental outputs. The most important of these is the surface finish. It is apparent from the test analysis that the frequency of vibration and the two force components hold a direct relation to the condition of the work piece material after cutting. While the degree of change feed, speed and depth of cut, will ultimately effect the surface finish, the variety of cutting tool will also impact the quality of the machined surface. In a quantitative analysis of the cutting conditions, with respect to the tool type, a direct correlation was found between the trends of both models. The tools exhibited similar characteristics, but a relation was found between them, relating to maximum efficiency machining. This portrayed the most efficient life cycle for the finer tool, and likewise for the medium cutting tool. In direct comparison, the forces and frequency that a tool is being subject to, is directly related to the power consumption of the lathe. The charts relating to surface finish, in particular, Ra, show the trends of wear for each tool and in particular there reliance over the chosen speed, feed, and depth of cut.

The Qualitative and quantitative models have been proposed, and with reference to literature, the models show very accurate conformity. Given this the neural network will now be implemented in attempt to predict these trends into an online system.

Throughout the qualitative and quantitative trends, all the conformance charts verified what had already been suggested in existing literature. Given that all the experimental trends conformed to those already established, it could be assumed that the experimental results are highly reliable. Given that the experimental results are

highly reliable, it suggests that the experimental setup and the experimental process were highly accurate, and reliable. The accuracy of the models reduces to the high degree of reliably repetitive results.

CHAPTER 7

ARTIFICIAL NEURAL NETWORK MODELS FOR PERFORMANCE PREDICTION

This chapter highlights the capabilities of the neural network back propagation algorithm in predicting performance variables. There is one particular model with three individual stages that will be tested for performance predictive capabilities. These three stages are illustrated in figure 7.1.

7.1 BACK PROPAGATION NEURAL NETWORK

The architecture with two layers back Propagation has been discussed extensively in chapters 3 and 4. This type of neural network was applied for three iterations. Figure 7.1 displays the three stages of discussion.

The first stage displays the four experimental variables as the only inputs. These variables being, the tool geometry, feed, speed and depth of cut. The outputs for the first stage will be the two experimentally measured force components. The BPNN will be trained and tested and a neural prediction of the quantitative force components will be given. This estimation will be compared with the actual experimental results, and the conformance of the results will be tabulated. Given the experimental results are considered to be the benchmark values, a percentage deviation will be shown with respect to the neural model.

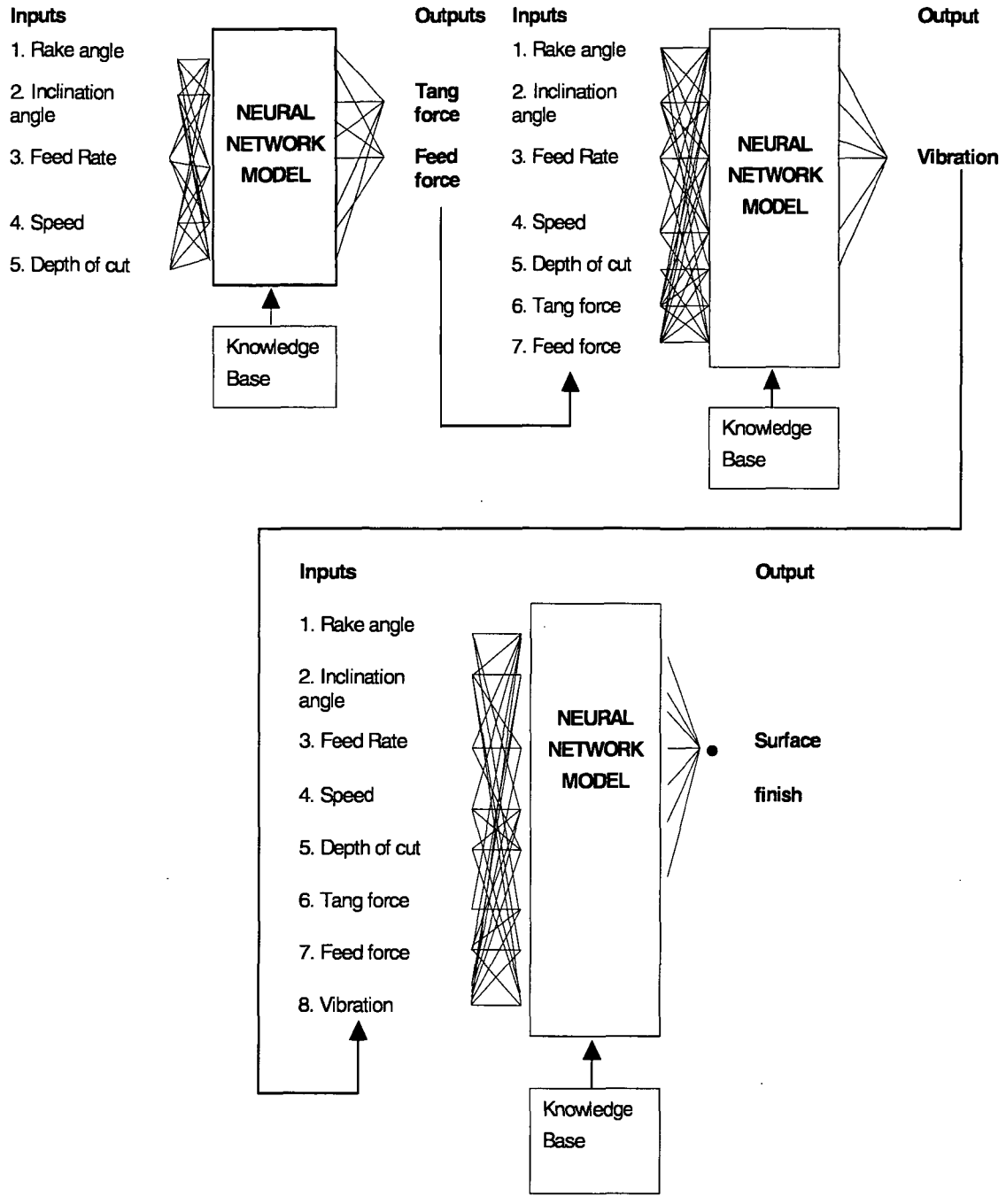

Figure 7.1 Neural network stages

The second stage follows the same trend, but given that the first stage portrayed a good correlation, the force components will now become inputs for the prediction of the frequency of vibration. Once again, if this stage displays accurate trends, the vibration will also become an input variable, and the surface quality will become the

output target value. Given there is a good correlation between the first two stages, a good correlation is expected in the third model, due to the increased number of quantitatively reliable inputs.

The implementation of a reliable source of online monitoring of the surface quality of a machined component, is the main goal of this field of work. Given that the third stage BPNN model prediction and experimental test data compare well, then the project goal will be achieved.

7.2 TRAINING THE BPNN ALGORITHM

The neural network model requires quantitative data to be able to form a trend model, before it can undertake testing. The initial test sample points are broken into two sections. The first section is of 194 data points, which are used for training purposes. The remaining 34 data points are used as a model for testing and comparison. The training procedure for a back propagation network is written as follows (based on figure 7.2) [75]:

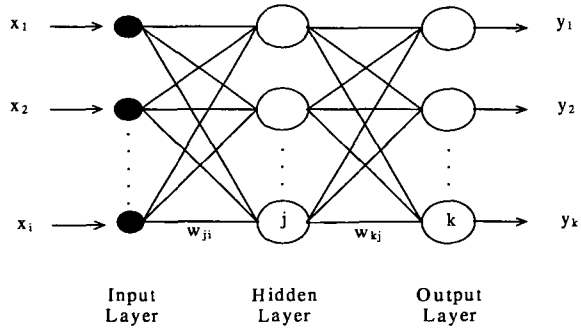

Figure 7.2 Back propagation neural network architecture

Step 1. Initialise the weights of the network at small random values.

Step 2. Start the learning cycle by exposing the network to a certain input pattern paired with the desired output.

Step 3. Compute the network's output, and compare it with the desired output so that the error can be calculated for the output and hidden layers, respectively.

Step 4. Adjust the weights of the network using the error back propagation algorithm so that a certain amount of the detected error is removed, if a momentum term is used.

Step 5. Repeat Steps 2 to 4 with all the input patterns and their correspondent desired outputs (training examples). Compute the cumulative error.

Step 6. If the cumulative error is within a tolerable range, terminate the training process, otherwise, go back to Step 2.

Once training is completed the testing phase can begin.

7.3 TESTING THE BPNN ALGORITHM

As mentioned in the previous section, 34 random data sample ranges were chosen for the testing stage. During the training phase, the neural network establishes a trend pattern based on the 194 sample ranges. This is by using the output variable as a guide. During testing, the neural model uses the obtained trend, along with the input variables, to approximate the output variable values.

It is important to note that the neural network model requires normalised data. This implies that the maximum value for each variable becomes equal to 1, and the minimum value becomes 0. Therefore all other data points exist between 0 and 1. This can be seen in the correlation between charts in appendix E. After the data has been normalised the BPNN is run. The results from the BPNN prediction for all three test stages, are displayed in the following sections.

7.4 BPNN FORCE PREDICTION STAGE

As explained previously three stages were involved in the prediction of online surface quality. The first stage, is the prediction of the force components. As seen in figure 7.1, the two force components, are the network outputs, with the tool type, speed, feed and depth of cut being the inputs. The quantitative model is made up of a training phase, and a testing phase.

As will be the case for all three neural models, there is going to be some non-conformance between the actual test results and the predicted model. This is summarised in the percentage error charts. The percentage error is calculated by:

$$\% \, Error = \frac{T\arg et - \Pr edicted}{T\arg et} \times 100 \qquad [7.1]$$

The full range of errors experienced can be seen in the related appendix for each section. The charts shown below simply illustrate the number of occurrences of each range of errors.

7.4.1 Training the BPNN for Force Prediction

The entire testing data displayed from the neural network prediction, can be viewed in appendix E. Figure 7.3 and figure 7.5 show the correlation between the experimental results (blue), and the BPNN predicted model (pink).

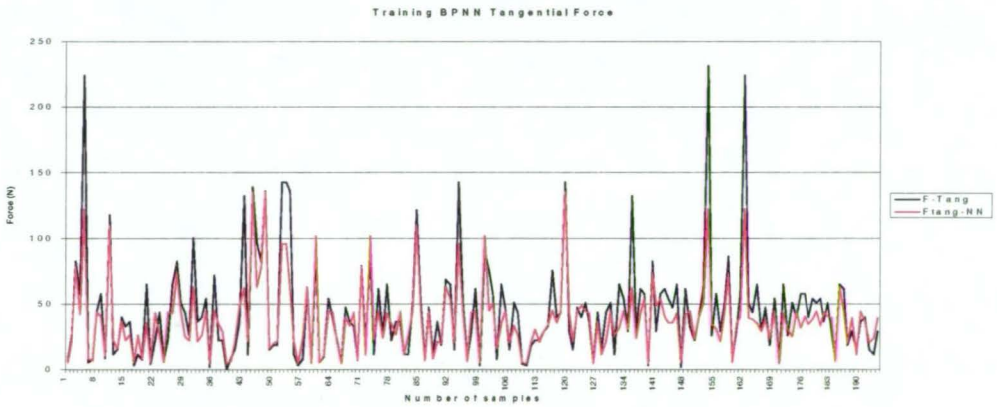

Figure 7.3 BPNN training prediction for tangential force

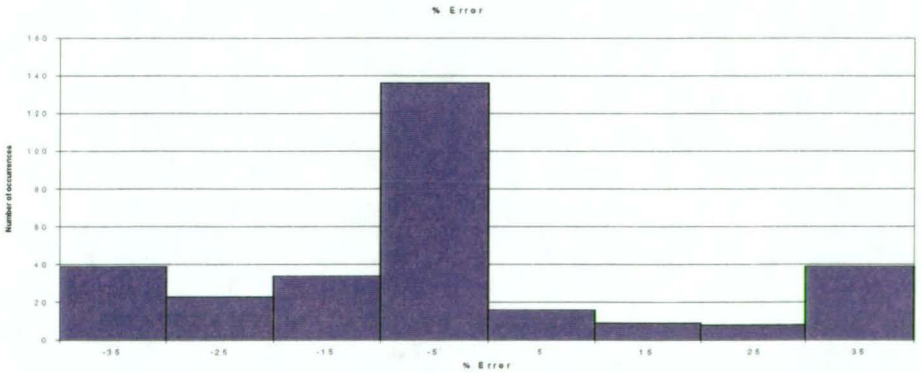

Figure 7.4 Occurrence of error for tangential force BPNN prediction

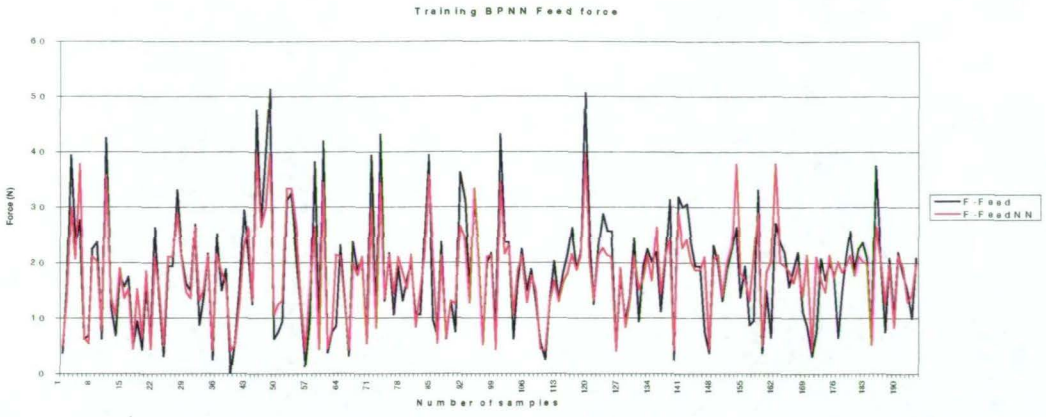

Figure 7.5 BPNN training prediction for feed force

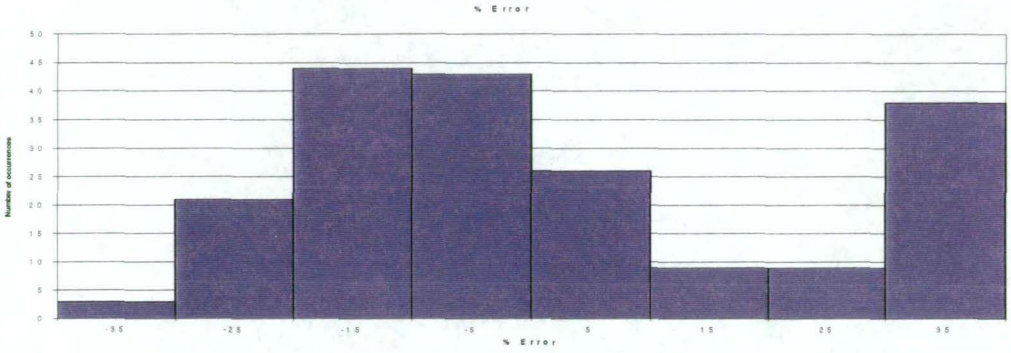

Figure 7.6 Occurrence of error for feed force BPNN prediction

Figure 7.1 shows the neural network model for this first iteration. The training of the BPNN for force estimation consisted of 4 input nodes, 10 hidden nodes and 2 output nodes for the structure. The four input nodes were tool geometry, feed, speed and depth of cut. As previously explained the data was normalised between zero and one.

Figure 7.3 shows how precisely the BPNN model conforms to the actual tangential force results. For 196 test samples there is very little deviation from the actual trend. This is further proven in figure 7.4, where the percentage error is dominated by errors falling between –10% and 10%. Given that the tangential force is almost precisely predicted by the BPNN model, the expected final outcome is going to be quantitatively reliable as the predictive importance of the tangential force is very high (11.6%).

Figure 7.5 portrays a similar model to that of the tangential force. The feed force training data follows the experimental results reasonably well, although it can be seen in figure 7.6 that the error ranges are slightly more scattered than for the tangential force. However based on the complete data (appendix E), there are very few select data points that don't actually conform with the predicted trend, which is also highlighted in figure 7.5.

Given that this training data is deemed accurate, it is feasible to continue with the testing phase.

7.4.2 Testing the BPNN for Force Prediction

Figures 7.7-7.10 displays the BPNN test results, and the percentage errors associated with the models predictions.

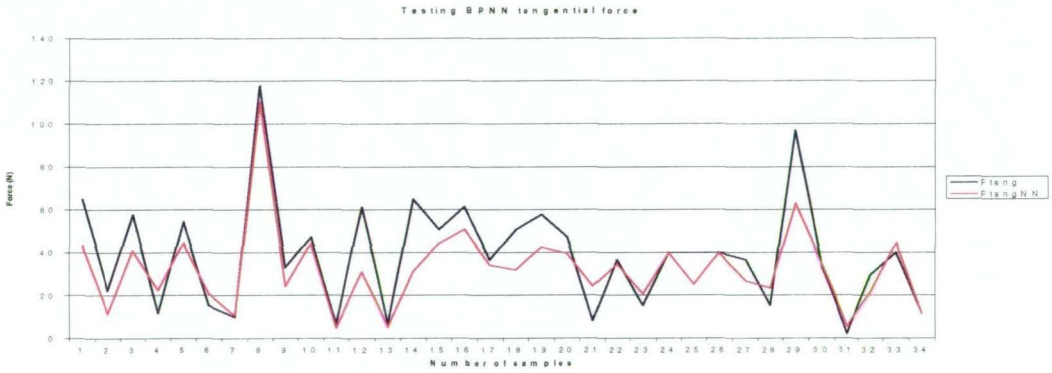

Figure 7.7 BPNN testing prediction for tangential force

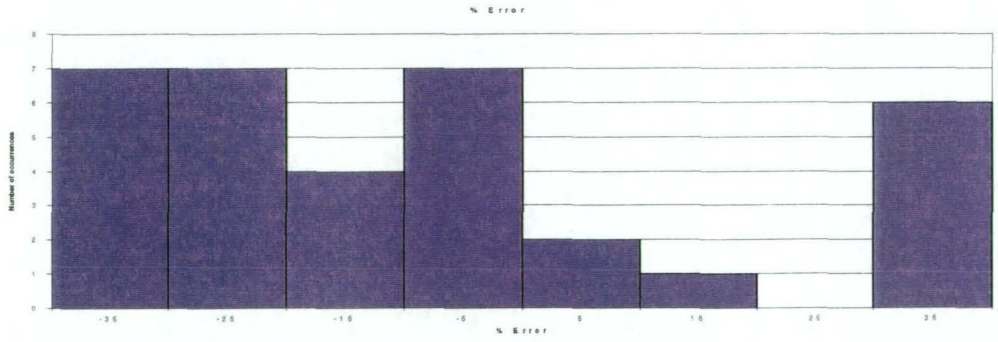

Figure 7.8 Occurrence of error for tangential force BPNN prediction

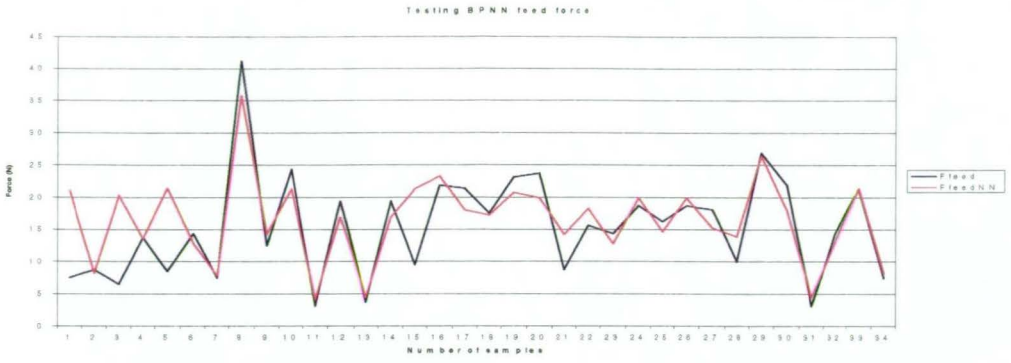

Figure 7.9 BPNN testing prediction for feed force

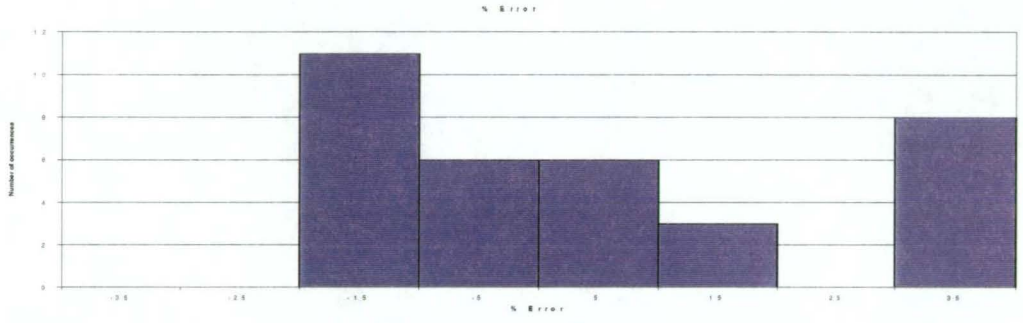

Figure 7.10 Occurrence of error for feed force BPNN prediction

For the tangential force neural network prediction it can be seen that the trend quite accurately follows the experimental results. It is shown that there is quite a scatter in the error ranges (figure 7.8), but these are mainly due to such data points with values of 14, 19, 25 and 29, where the BPNN model predicts the trend well, but does not conform to the actual value. Apart from these discrepancies the model is almost a precise approximation to the experimental data.

The feed force once again highlights the results obtained with the tangential force. The BPNN model follows the trend of the experimental data almost precisely. It is only for the first five sample points where there is a reasonable deviation in the conformance of the predicted and actual data. This is shown in figure 7.10, where small discrepancies in the order of +/- 20 % (or less) dominate. The large degree of errors in the range between 20% and 30% are large only because of the first five data points. Apart form these the BPNN predicts the experimental trends with precise accuracy.

7.5 BPNN VIBRATION PREDICTION STAGE

Given that the first stage of the neural network implementation was completed successfully and with very little error, it is feasible to increase the number of input variables and continue with the next stage of prediction. There are now 6 input nodes, 10 hidden nodes, and 1 output node. The input nodes remain the same as previously, but with the addition of the two force components (proven to be valid inputs). The frequency of vibration now becomes the output target value. The sequential methodology is displayed in figure 7.1.

7.5.1 Training the BPNN for Vibration Prediction

The following charts display the BPNN predictions for the frequency of vibration with the increased input variables for training.

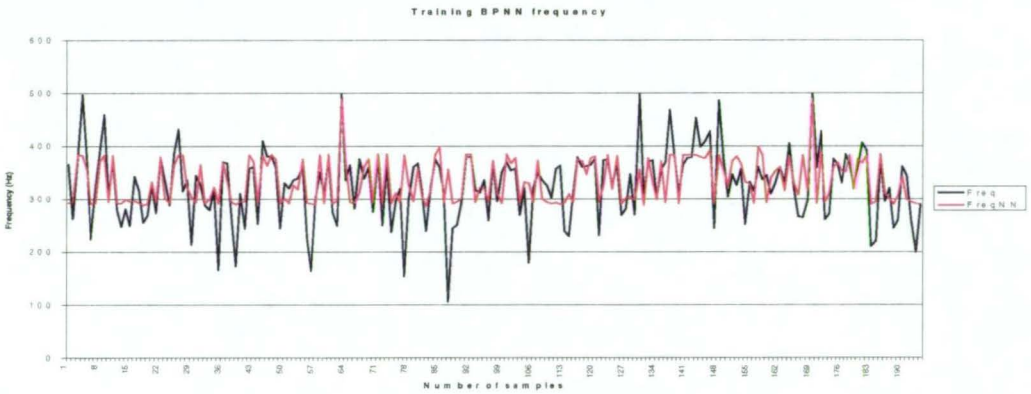

Figure 7.11 BPNN training prediction for vibration

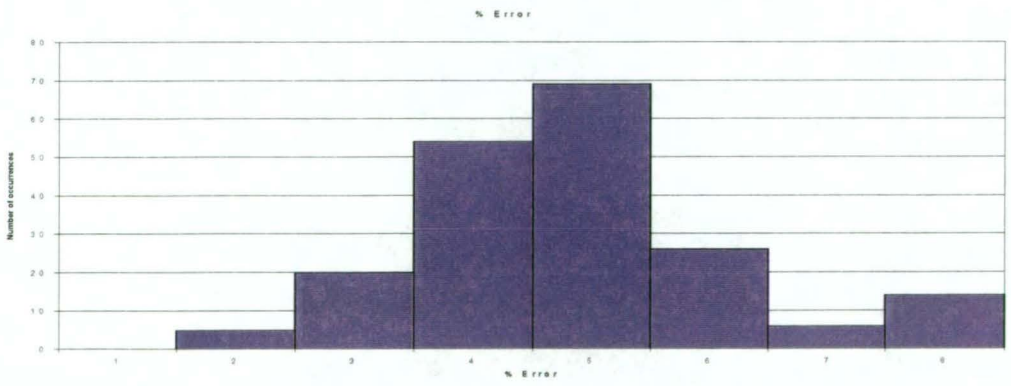

Figure 7.12 Occurrence of error for vibration BPNN prediction

Once again the training procedure for the neural network is completed successfully. The conformance between the actual and predicted data is very accurate, which is also highlighted in figure 7.11. There is shown to be very little error (figure 7.12) greater than +/-10 %, which displays an almost flawless implementation of the neural network model. The only noticeable variance is for large maximum and minimum peaks. Overall the BPNN model is able to estimate these values to complete precision and any small inconsistencies could be due to the number of iterations involved for algorithm convergence. This particular BPNN model was trained over 1000 iterations, given a larger iterative process, these peaks could have been achieved.

7.5.2 Testing the BPNN for Vibration Prediction

Figure 7.13 displays the neural network models prediction based on the 34 testing data points.

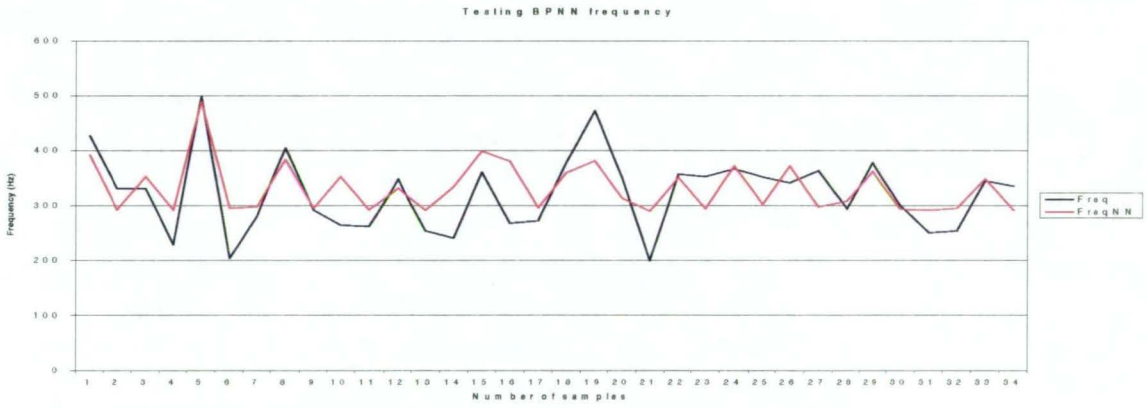

Figure 7.13 BPNN testing prediction for vibration

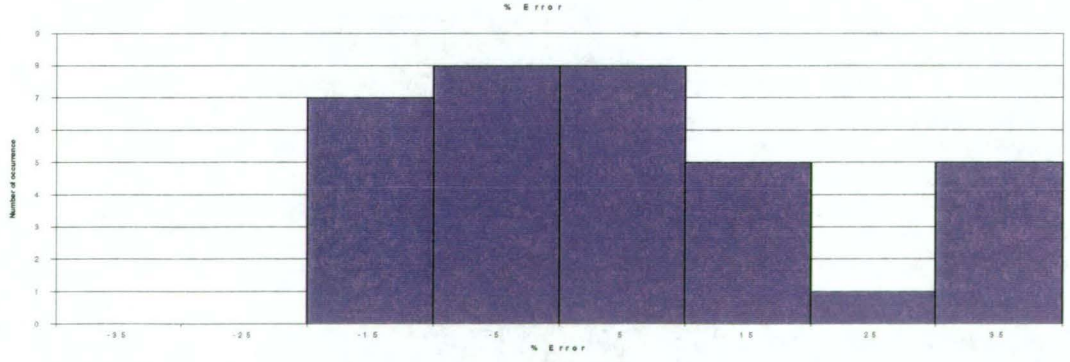

Figure 7.14 Occurrence of error for vibration BPNN prediction

While the two trend lines in figure 7.13 don't coordinate completely with one another, the error analysis in figure 7.14 shows that there is actually only minimal errors between actual and predicted values, hence there is no bias in the error for prediction. The quantitative values are in Hz, and the ability of the BPNN to predict to such an accurate level for a dynamic system is found to be highly satisfactory. This tells us that neural network prediction, although not visually conforming, actually gives a very good estimation to the actual experimental values. Given that the error percentage is dominated by low values left and right of zero, it displays uniformity in the error analysis.

7.6 BPNN SURFACE FINISH PREDICTION STAGE

By estimating vibrations through the BPNN, the second stage of the process was completed. Based on figure 7.1, the first two stages of the process were completed successfully making it feasible to iterate the third stage. This stage is of the highest priority and given the successful implementation, it will pave the way for an online based surface monitoring system. This result would be expected as the first stage, using only 65.7% predictive importance, was extremely successful. Also the second stage, with 88.4% predictive importance, was implemented successfully. Now 100% of the predictive importance variables are in use. This means the neural network will be trained and tested with its maximum possible efficiency.

7.6.1 Training the BPNN for Surface Finish Prediction

Figure 7.15 shows just how accurate the predicted model is with reference to the experimental values.

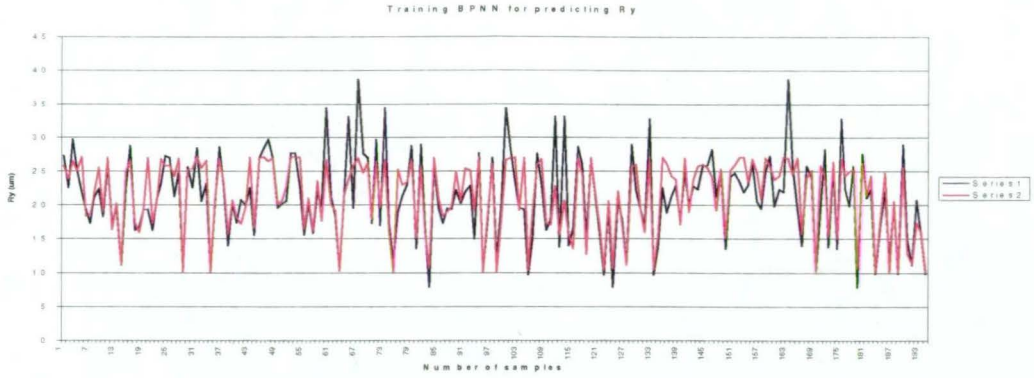

Figure 7.15 BPNN training prediction for surface finish

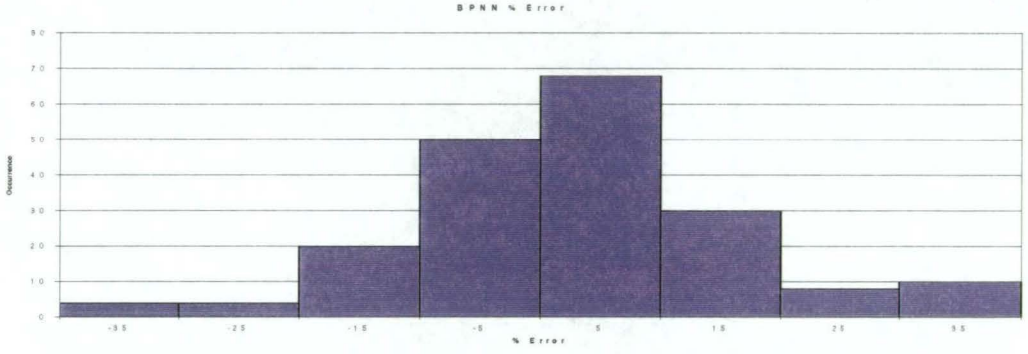

Figure 7.16 Occurrence of error for surface finish BPNN prediction

As expected the model predicts the accuracy almost precisely to that of the experimental data. The average error is only 3% of the target value. This is for a length value that is already in micro metres, making the discrepancy almost negligible. This is displayed in figure 7.16, were basically no real error exists outside the middle ranges. This proves that with all the predictive performance values incorporated into the BPNN model, and with such a high precision testing process, the neural network function gives a highly organised training pattern, that will ensure an accurate testing process.

7.6.2 Testing the BPNN Surface Finish Prediction

The BPNN model has been trained to the highest degree of accuracy. This poses the question, can the neural network model test the data points with enough accuracy to give a reliable model? Figure 7.17 and figure 7.18 display the answer to this question

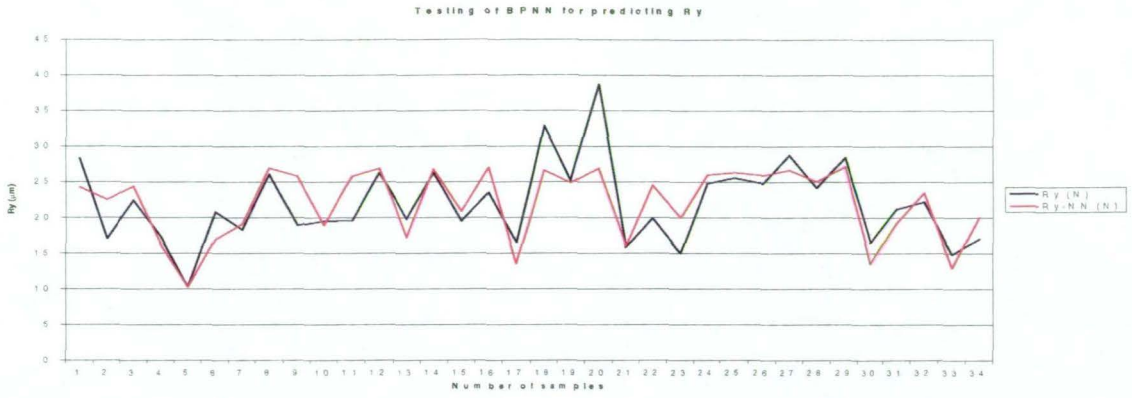

Figure 7.17 BPNN testing prediction for surface finish

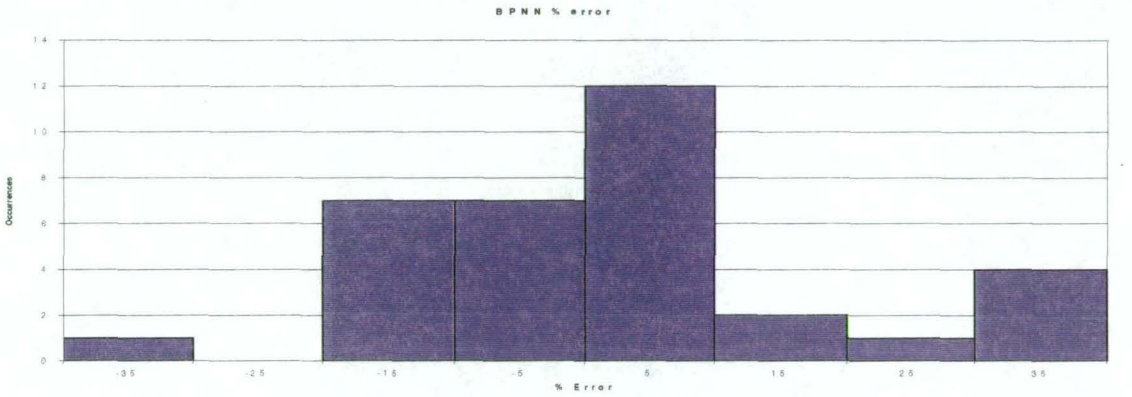

Figure 7.18 Occurrence of error for surface finish BPNN model

The average error associated with this final testing phase is 1.9%. This is an extraordinary result given the fact that the neural network model is approximating values in the range of 10^{-6} m. Figure 7.17 verifies the accuracy of the model, whereby the predicted and experimental results almost conform perfectly along the same sets of trend lines. Figure 7.18, also shows the high level of minimal errors, as opposed to larger error ranges.

This result is above all expectations. The accuracy achieved is a result of:

1. Well calibrated testing equipment
2. A comprehensive range of testing
3. A very reliable neural network

This result guarantees, that for any particular variable cutting condition, the surface finish can be accurately determined to within an average of 1.9% of the actual value. Incorporated online, and utilising a BPNN, this model can determine the surface quality of any component to the highest degree of precision.

7.7 PREDICTIVE IMPORTANCE

It is important to discuss the predictive importance of the neural network model at this point, to explain why the three-stage model was chosen, and also to clarify the increasing accuracy throughout the later stages of prediction. Before the initial model begins its training phase it will predict how influential each of the seven test variables is on the final output.

Predictive importance is concerned with the increase in generalisation error when a variable is omitted from the training set. An analysis using predictive importance is completed by initially training the network using all the inputs, *i*. The network is then retrained with a single input omitted from the model to study the change in the network error, hence, *i-1* inputs are used. This is completed *i* times with a different input omitted in each instance. The resulting change in error in each instance is a direct measure of the predictive importance. An increase in error indicates the omitted input is adding values to the networks decision, while a decrease, or no change, indicates the omitted input was not contributing to the networks prediction. Predictive importance is the most suitable measure of input variable importance in applications where a neural network is used for predictive modeling, trained using observational data where the training cases are a random sample taken from some population.

The predictive model for this process incorporated the effect of:

1 Feed rate
2 Speed rate
3 Depth of cut
4 Tool geometry

5 Tangential force

6 Feed force

7 Vibration

On the resulting output, the surface finish. Table 7.1 shows the quantitative reliance the neural model has with regard to each of these variables.

Ranking	Parameter Name	Percentage Contribution (%)
1	Feed	34.4
2	Speed	12.4
3	Tang.F	11.6
4	Freq	11.5
5	Feed.F	11.1
6	Depth	10.4
7	Tool	8.5
	sum	100.0

Table 7.1 Predictive importance of BPNN inputs

What the predictive importance table demonstrates, is that the feed rate has the highest contribution towards establishing the neural network model for surface finish quality. Likewise the type and geometry of the tool used, has the least impact on the quantitative model. While predictive importance is only considered a qualitative indication, this method is more appropriate in excluding "non-contributing" parameters in the model. This will be of great significance when numerous parameters are involved in the process modeling.

7.8 CONCLUDING REMARKS

It can be seen from the accuracy of the neural network model that it is a reliable quantitative method of predicting machining parameters. It has been seen that mechanics of cutting approach and the empirical approach have been used for the prediction of performance variables in direct modeling situations and have had some

success with an accuracy of +/-15% error range. The problem with the mechanics of cutting models is the inability to predict all experimental conditions, while the empirical approach requires extensive experimentation for a questionable quality of accuracy.

The current investigation, using three stages of development, was proposed to alleviate the above problem. The three stages were incorporated so that the predictive importance variables would be increased after each prediction. Given that the final prediction utilised 100% of the predictive importance, and given the first two models were successful, a reliably quantitative model would be expected in the third iteration.

The neural network model, in particular a back propagation system, proved to be far superior to any method currently investigated. The breakthrough for industry of being able to achieve an average error of 1.9% gives them a new alternative to quality control. Not only can they now function within a certain error range, they can accurately work to an almost precise performance value. To be able to pinpoint the quality of the surface finish (measured in micro metres), to within a 1.9% error margin, improves the quality manufacturing approach tremendously, and when applying with automated techniques, introduces the concept of effectively removing "down" time, and proportionally saving the industry precious time and money.

CHAPTER 8

FINAL CONCLUDING REMARKS AND PROPOSED FUTURE WORK

8.1 FINAL CONCLUDING REMARKS

In this thesis a need for quantitative machining in all facets of the manufacturing environment is highlighted. In a world were the demand is for low cost high quality products, there is no longer any margin for shortcomings or errors. In the past a "just" product, produced by countless staff over an unconcerning time period, would have sufficed. At the end of the day, the product was sold, the employees were paid, and the executives were happy. Today's manufacturing environment has no remnants of these former trends. The product must be produced to the highest quality specifications, and must be highly marketable. Therefore it must be a good product, that is premium value for money. For this to be the case, the manufacturing process used to generate the product must be optimised.

It is shown that with regard to a quality product, the main issue in metal machining is the quality of the output components surface finish. Each individual type of manufacturing industry will have some of conformance standard or code, to which the surface finish of all produced components must comply. In particular, in metal turning the surface quality was depicted by surface deviation distance. If this deviation falls below the company's standards (usually 5%), then the machined component is deemed unsuitable for further processing and will be re-melted or scrapped. While this can be very costly and time consuming, so to are the current processes to determine the actual state of the surface finish.

This problem has been tackled through various techniques. The most basic of these is the fundamental mechanics cutting approach. Researchers in the field of classic

orthogonal or oblique turning, have devised an extensive set of equations relating the forces and power in the system to the main turning variables, in speed, feed and depth of cut. While these equations model the "ideal' force reasonable well, the impact of the actual experimental variables cause these approximations to be unrealistic. These methods have been shown to be of accuracy of +/- 15%. Figuratively speaking it is not viable to incorporate the mechanics of cutting approach into the production process when no real accurate conclusion can drawn as to the state of the work piece material.

An advancement of this technique is an empirical model. This represents data that has been experimentally established, and tabulated. Since it is based on actual machining conditions, the quality of the empirical model is largely based on the number of process variables. The more variables that are incorporated into the process, the more accurate the actual prediction of the machining status will be. While the empirical methods realise the importance of online conditions, they can not possibly monitor changes within the machining variables. Given that these variables are changing rapidly, the empirical approach offers little satisfaction towards a premium solution.

It is clear that that the machining variables are completely unpredictable for a text based approach. There is a need to have some relevance to the actual process, to feasibly analyse the systems performance. Investigation has been done into this area through both offline and online methods.

Offline techniques deal with the current analytical methods that are used in industry today. After a component has been machined, the surface quality is checked via some technique, and a decision is made to the quality of that component. These processes deal with geometric properties, whether through wear maps, microscopy, or geometric modeling. The main issue is, as reliable as these methods may be, the time taken for a decision on the product conformance, is time lost. Furthermore, after the test has been completed, the machining variables would have already changed, and therefore repeatability of the process will lack any form of quantitative reliability. This strengthens the fact, that only online testing techniques will

accurately and reliably predict the quality of the machining process, in a quick and repeatable manner.

Extensive investigation has been done into online surface wear determination, but to this stage a reliably predictive model is yet to be fully developed. Basic experimentation into force, vibration and surface finish properties of the online system have done little more than add to the empirical approach. The major breakthrough in online testing has come with implementation of online smart systems. The most efficient of these systems is the neural network methodology. In conjunction with extensive experimental data, the neural network is able to accurately predict the trends of the machining process. Most importantly, the neural network is able to recognise that the machining variables are constantly changing, and formulate reliably quantitative prediction based upon the changes. For a neural network to accurately predict an outcome, it needs extensive training. This is the main set back of the current investigations into the neural network anthology. The studies have restricted the number of machining variables used for training the network model, and hence the neural network can not make a reliable prediction to the actual level of the surface quality being achieved.

An extensive experimental investigation covering 10 feed rates, 11 depth of cut, 15 speeds and varieties of tools, were carried out over a range of 240 cuts. These cuts occurred over test samples of 20cm in length and 8cm diameter. Two different cutting tools were used to examine the reliability of the neural network prediction on surface finish. For a more advanced quantitative analysis, further investigation should be undertaken in this field. Data acquisition occurred through Labviews new G-code based interface. The data acquisition was set up to record the extensively calibrated force components, and the frequency of vibration. A fully functional program was written in this code to enable the data acquisition process to occur. This program code allowed for the automatic retrieval of variable measurements through a digital interface. Furthermore a surface analyser was used to record two components of the surface profile. All these machining outputs become training data for the neural network model. Thus, rather than restricting the neural network, it allows for maximum efficiency when creating a prediction at online surface quality.

This test data forms the basis of a study. Qualitative and quantitative trends help compliment the published results in literature, and help verify the quantitative reliability of the data acquisition equipment.

By complimenting previous findings, the force components of the turning system increased with increasing feed and depth of cut, but were largely unaffected by the varying speed rates. The frequency of vibration also increased with increasing feed and depth of cut, but decreased as the speed rate was increased. The surface roughness decreased in quality as the depth of cut and feed were increased, but as the speed rate was increased the surface profile improved. All these trends conform with previous findings and suggest a very stable and precise experimental base.

In studying the quantitative effects of the increase in hardness of the cutting tool, the same trends were generated as for the qualitative assessment, however noticeable differences were generated between the two tool types. As expected for low feed, speed and depth of cut, the softer tool had the higher forces and frequency of vibration, yet achieved a better surface quality than the harder tool. At larger variable rates, the softer tool became unstable and the surface quality deteriorated at a far greater rate than the harder tool. This meant that after certain cutting conditions the harder tool would generate a better quality in the surface conditions, while consuming less power to achieve the desired result. While literature proposed these results, the quantitative analysis provided information on the critical changeover points required to maximise the efficiency of the production process.

Neural network architecture was developed to use as both a direct and inverse model for performance prediction. A multi-layered perception with back propagation was used for this purpose. The network is essentially used as three stages where in the first stage the four basic cutting variables are used to predict the two output components, being the feed force and the tangential force. Given the successful implementation of the first stage the second stage adds the two force components to its inputs, to predict the frequency of vibration. Similarly if this quantitative prediction was successful, the third stage is implemented. The successful implementation through training and testing of the vibration model, ensures the third stage was to be a reliable prediction. This occurs with all seven input nodes, ten

hidden nodes and one output node, being the surface finish. All seven inputs have a degree of predictive importance, or influence on the predictive process of the neural network. Given that each stage has more added inputs, the final execution is bound to be the most quantitatively reliable as all the predictive importance (100%) factors are incorporated. The model for surface finish shows one of the closest conformities to be seen in most modern neural network predictions. The final model proposed an average operating error of 1.9% error. This is in contrast to current industry operating conditions where predictive error yields +/-15%.Even more astounding is the ability to predict a quantity in micro metres to within 1.9% of the actual value. This result reflects that all predictions made in the qualitative and quantitative scale are indeed very accurate.

The ability to predict the surface finish online reliably has been a major goal for researchers and industry. While methods have been developed, the actual predictive efficiency has been quantitatively unreliable. The importance to industry of a predictive method obtaining accuracy to within 1.9% of the exact value is a major breakthrough, and offers the potential to relieve wastage, re-melting, re-cutting and most importantly, money. So where does this field of study go from here?

8.2 PROPOSED FUTURE WORK

This work develops a neural network model for predicting the online quality of surface roughness. Given that all qualitative and quantitative models conform to literature, and the multi layered neural network model predicts the surface quality to a precise value, it can be considered that this investigation is thoroughly successful.

A logical development of this work is to incorporate the neural model online with the test setup, and enable a form of digital data display for online measurements. This will enable continuos monitoring of the surface finish profile, while machining continues without interruption. With some alteration of the neural network source code, to incorporate the manufacturers conformance standards, the actual manufacturing process could be stopped or halted when the surface profile becomes out of range. Given that surface quality is almost directly related to the cutting tool

condition or quality, the model can be incorporated with existing technology, such as automatic tool changes. This proposes that at the instant the surface finish is no longer considered within quality standards, the condition will be monitored and the necessary changes made, to ensure a continuos flow of machining. This gives manufacturing the edge it has been seeking.

REFERENCES

1. VISHY KARRI, and M. GILLIES, "Automatic tool selection using Neural Networks" Abstract paper, Dept of Mechanical Engineering Tasmania, 1995.

2. AMITABHA BHATTACHARYYA, and INYONG HAM, "Design of Cutting Tools, use of metal cutting theory", American Society of Tool and Manufacturing Engineers, 1969.

3. E. J.ARMEREGO, "Material Removal Processes, an Intermediate Course", Manufacturing science group, 1993.

4. MAX KURREIN, and F.C LEA, "Cutting Tools for Metal Machining", Charles Griffin & Company, 1940.

5. RICHARD R. KIBBE, JOHN E. NEELY, ROLAND O. MEYER, and WARREN T. WHITE, "Machine Tool Practices", John Wiley & Sons, 1979.

6. VISHY KARRI, Manufacturing course notes, "Surface Finish", Dept of Mechanical Engineering Tasmania, 1997, pp 66-67.

7. MCLEONG, "Statistical Process Control for Tools", http://www.info-acess.com/Netizens/Mcleong/tool.htm

8. YELLOWLEY. I and LAI.C.T, "The use of Force Ratios in the Tracking of tool wear in Turning", ASME, J. Eng. For Ind, 115, 1993, pp 370-373.

9. ELANAYAR, S.V.T. and SHIN, Y.C., "Modeling of Tool Forces for Worn Tools: - Flank Wear Effects", Materials Issues in Machining II and The Physics of Machining Processes II, TMS & ASME, pp. 341-361, Symposium on "Mechanics of Machining", ASME WAM, 1994.

10. G.M. ZHONG, and T.W. HWANG, "Analysis of Cutting Dynamics in Microscale", ASME, 1990.

11. S.G. KAPOR, and R.E. DEVOR, "Force Modelling for Worn Cutting Tools", Thesis Abstract, 1996, FY97 MTAMRI.

12. C.Y.H. LIM, S.C. LIM, and K.S. LEE, "Wear of TiC-coated carbide tools in dry turning", Wear, Vol. 225-229, 1999, pp. 354-367.

13. E. WASCHKIES. C. SKLARCZYK. and K. HEPP, "Tool Wear Monitoring in Turning", JEI, 1994, (116:4), pp 521, Institute of non destructive testing. Germany.

14. FABIO NOGUEIRA LEAO, "Tool Life Monitoring through Vibration Signal and Motor Current in Turning Process", ASME journal 1991.

15. S.CHO, and K.KOMVOPOULOS, "Correlation Between Acoustic Emission and Wear of Multi-Layered tools", 2nd issue of the ASME Journal of Manufacturing Science and Technology, Ca 94720.

16. J.A.RICE, S.M.REID, and MEAM, "Feasibility of Catastrophic Cutting Tool Fracture Prediction via Acoustic Emission Analysis". Department of Mechanical Engineering Chicago. Department of Engineering Uni of Michigan Ann Arbor.

17. A.A HOOD, "Control System Design for Active Vibration control of a Turning Process using PMN Actuators". MS 96-13 1996.

18. E. MARUI, M. HASHIMOTO, and S KATO, "Damping Capacity of Turning Tools PT1. Effect of Clamping Conditions and Optimum Clamping load". Department of Mech Eng, Gifu Uni, Japan, Dept of Mech Eng, Toyota College of Technology, Japan.

19. E. MARUI, M. HASHIMOTO, and S KATO, "Damping Capacity of Turning Tools PT2. Mechanisms Initiating Damping Capacity". Department of Mech Eng, Gifu Uni, Japan, Dept of Mech Eng, Toyota College of Technology, Japan.

20. KAI FRANK GOEBEL, "Management of Uncertain in Sensor Validation, Sensor Fusion, Diagnosis of Mechanical Systems using Soft Computing Techniques". Dept Mech Eng, University of California, Berkely.

21. X. D. FANG, "In Process Evaluation of the Overall Machining Performance in Finish-Turning, Single Data Source". Dept Mech Eng. Wollongong NSW.

22. ASME Journal of Manufacturing Science and Technology. Issue 3.

23. SHIN, Y.C. AND OH, S.J, "Surface Roughness Measurement by Ultrasonic Sensing for In-process Monitoring", ASME DSC-Vol. 50, Symp. on Computer Control Machines for Manufacturing, Dec. 1993, pp. 3-12, New Orleans, LA.

24. J. COLGAN, H.C CHIN, K. DANAI, and S. R. HAYASHI, "On Line Tool Breakage Detection-A Multi Sensor Method", Dept Mech Eng, Massachusetts., Corp Research and Development, Schenectady NY.

25. T. IHARA, "Detection of Control of Excess Vibration, Tool Breakage and Tool Wear". N.Moriakj.. IN AP 90, p13.

26. ARMERAGO, E.J.A., "Practical Implications of Classic Thin Shera Zone Analysis", UNESCO/CIRP Seminar on Manufacturing Technology., Singapore, 1982, Pg167.

27. KARRI,V, Fundamental Studies of Rotary Tool Cutting processes, Ph.D Thesis, The university of Melbourne, 1991.

28. ARMERAGO, E.J.A., and BROWN, R.H., "The Machining of Metals", Prentice Hall inc, New Jersey, 1969

29. ARMERAGO, E.J.A and DESHPANDE, N.P, "Computerised Predictive Cutting Models for Forces in End Milling Including Eccentricity Effects", Annals CIRP, 1989, Vol.38/1,p.45.

30. KARRI, V., "A Neural Network controller in Automated Wood Machining". IASTED Int. Conference On Robotics and Manufacturing, Honolulu, USA. Aug. 19-22, 247-251. 1996.

31. www.whatis.com/neural networks, "neural networks", What is.com, February 1999.

32. CHOON SEONG LEEM, D. A. DORNFIELD, and S. E. DREYFUS, "Customised Neural Network for Sensor Fusion of Acoustic Emission and Force in Online Detection of Tool Wear".

33. BARSHDORFF, and D. FEMMER, "Artificial Neural Networks for Wear Estimation", Institute of Electrical Measurement, University of Paderborn.

34. BARSHDORFF. D, MONOSTORI. L, KOTTENSTEDE. T, WARNECKE, G, and MULLER, M. (1993a). "Cutting tool monitoring in turning under varying cutting conditions; An artificial neural network approach". The sixth International conference on Industrial and Engineering Applications of Artificial Intelligence & Expert Systems.

35. BARSHDORFF. D, MONOSTORI. L., KOTTENSTEDE. T, WARNECKE, G, and MULLER, M, (1993b). "Wear Estimation and state classification of cutting tools in turning via artificial neural networks", Proc. Of Tooldiag `93, Intern. Conf. on Fault Diagnosis.

36. BARSHDORFF. D, and BOTHE. A, (1991). "Signal Classification Using a New Self-Organising and Fast Converging Neural Network". Noise and Vibration Worldwide, Elsevier Advanced Technology, pp 11-19.

37. BARSHDORFF. D, MONOSTORI. L, (1991). "Neural Networks, their application and perspectives in the intelligent machining". In: Computers in Industry, 17, Elsevier Science Publishers B.V., pp 101-119.

38. DEVIJER. P.A, and KITTLER. A, (1982): "Pattern Recognition – a statistical approach", Prentice Hall, New Jersey.

39. KOHONEN, TEUVO. (1986). "Learning Vector Quantization for Pattern Recognition". Helsinki University of Technology, Laboratory of Computer and Information Science, Report TKK-F-A601

40. KRUGER. J, and SUWALSKI. I, (1992). "Fuzzy Logik and Neuronale Netz in der Maschinendiagnose". ZwF87, 11.

41. ROSENBERGER. U, and BAHRE. D, (1993). "Ablauf erkennen. Prozebidentifikation beim spanenden, Fertigen mit Hilfe neuronaler Netze", Maschinenmarkt, Wurzburg 99, pp. 50-54

42. RUMELHALT. D. E, and CLELAND. J. L, (1998), "Parallel Distributed Processing" Volume 1: Foundations, A Bradford Book.

43. SAMMON. JOHN. W. JR, (1969), "A nonlinear mapping for data structure analysis", IEEE Transactions on Computers, C-18(5): 401-409.

44. B. SICK, and A. SICHENEDER, "Time Delay Neural Networks for On-Line Tool Wear Classifications and Estimation in Turning", Proceedings of the Third Conference on Neural Networks, S461-466.

45. R. G. KHANCHUSTAMBHAM, and G. M. ZHANG, "A Neural Network Approach to On-Line Monitoring of Machining Processes", MS 92-5. 1992.

46. http://active.uuis.ca/~zbr/tool wear.html, March 1999.

47. "Technologies of Tool Wear Monitoring", Research project of the University of Berkely, Mech Engineering Department.

48. JONG JIN PARK, and A. GALIP ULSOY, "On Line Flank Wear Estimation using an Adaptive Observer and Vision. Part 1. Theory", Dept of Mech Eng, University of Michigan, Ann Arbor.

49. JONG JIN PARK, and A. GALIP ULSOY, "On Line Flank Wear Estimation using an Adaptive Observer and Vision. Part 2. Experimentation", Dept of Mech Eng, University of Michigan, Ann Arbor.

50. GALIP ULSOY, "Real Time Estimation of Tool Wear in Machining operations", Department of Mechanical Engineering, University of Michigan Ann Arbor.

51. www.mtamri.me.uiuc.edu/asme/old_papers/paper95/papers.html, February 1999.

52. www.mtamri.me.uiuc.edu/asme/old_papers/paper94/94paper, February 1999.

53. www.mtamri.me.uiuc.edu/asme/old_papers/paper93/papers.html, February 1999.

54. "Introduction to Material Science for Engineers", Chapter 4, 2nd Edition, Shackleford.

55. "Elements of Material Science and Engineering", Chapter 7, 6th Edition. Van Vlack.

56. "Principles of Material Science and Engineering", Chapter 4, 1st Edition. Smith.

57. Lawrence Livermore National Laboratory online. Operated by the University of California for the U.S. Department of Energy. "Radiography", 1999.

58. "A Magnetostrictive Actuator Based Micropositioner and its Application in Turning". Proc. Of SPIE Symposium on Smart Structures and Materials, Vol. 2721, March 1996, pp. 385-393.

59. www.whatis.com/neural/fuzzylogic, "fuzzy logic", What is.com, February 1999.

60. TAEJOKO, and DONG WOO CHO, "Tool Wear Monitoring in Diamond Turning Through fuzzy pattern recognition", 1991.

61. DR KHW SEAH, "In Process Tool Wear Monitoring", 1990.

62. www.eng.nus.edu.sg/Eresnews/May95/kslee.html, March 1999.

63. CAUDILL, M. and BUTLER, C., "Naturally Intelligent Systems", Massachusetts Institute of Technology, 1990.

64. CAUDILL, M. and BUTLER, C., "Understanding Neural Networks - Computer Explorations", vol. 1: Basic Networks, Massachusetts Institute of Technology, 1992.

65. HERTZ, J., KROGH, A. and PALMER, R. G., "Introduction to the Theory of Neural Computing", Addison-Wesley Publishing Company, 1991.

66. WEIJTERS, A. J. M. M. and HOPPENBROUWES, G. A. J., "Backpropagation Networks for Grapheme-Phoneme Conversion: a Non-Technical Introduction", Artificial Neural Networks, Springer-Verlag Berlin Heidelberg, 1995, pp. 13-36.

67. FREEMAN, J. A. and SKAPURA, D. M., "Neural Networks: Algorithms, Applications, and Programming Techniques", Addison-Wesley Publishing Company, Inc., 1992.

68. MASTERS, T., "Practical Neural Network Recipes in C++", Academic Press Inc., 1993.

69. BATTELLE, "Neural Networks: What are Neural Networks?", http://www.emsl. pnl.gov:2080/proj/neuron/neural/what.html, 11th Jun. 1998.

70. KANDIL, N., KHORASANI, K., PATEL, R. V. and SOOD, V. K., "Optimum Learning Rate for Backpropagation Neural Networks", Neural Networks: Theory, Technology and Applications, Institute of Electrical and Electronic Engineers, 1996.

71. RAO, D. H., GUPTA, M. M. and WOOD, H. C., "Neural Networks in Control Systems", Neural Networks: Theory, Technology and Applications, Institute of Electrical and Electronic Engineers, 1996.

72. CHOW, M., SHARPE, R. N. and HUNG, J. C., "On the Application and Design of Artificial Neural Networks for Motor Fault Detection - Part I", Institute of Electrical and Electronic Engineers Transactions on Industrial electronics, vol. 40, no. 2, 1993, pp.181-188.

73. WASSERMAN, P. D., "Advanced Methods in Neural Computing", Van Nostrand Reinhold, 1993.

74. HUANG, S. H. and ZHANG, H. C., "Artificial Neural Networks in Manufacturing: Concepts, Applications and Perspectives", Institute of Electrical and Electronic Engineers Transactions on Components, Packaging and Manufacturing Technology, Part A, vol. 17, no. 2, 1994, pp. 212-228.

75. RUMELHART, D. E. and McCLELLAND, J. L., "Parallel Distributed Processing", vol. 2, M. I. T. Press, Cambridge, MA, 1986.

76. LIPPMAN, R. P., "An Introduction to Computing with Neural Nets", Institute of Electrical and Electronic Engineers ASSP Mag., April 1987.

77. WILLIS, M. J., MONTAGUE, G. A. and PEEL, C., "On the Application of Artificial Neural Networks for Process Control", Kluwer Academic Publishers, London, 1995.

78. KARRI, V., "Practical Application of Neural Nets for Manufacturing Processes", EMEC 95, Auckland, Sept. 1995, pp.103-113.

79. TARNG, Y. S., HSEIB, Y. W. and HWANG, D., "Sensing Tool Breakage with Neural Network", Int. Mach. Tools Manuf., vol. 34, no. 3, 1994, pp.341-350.

80. BARSCHDORFF, D. and FEMMER, U., "Artificial Neural Networks for Wear Estimation", Int. Conf. on Int. Manuf.Sys., Vienna, June 13-15th,1994, pp.157-161.

81. BARSCHDORFF, D. and MONOSTORI, L., "Neural Networks: Their Application and Perspectives in Intelligent Machining", First CIRP Workshop of the Intelligent Manufacturing Systems Seminars on Learning in IMS, Budapest, Hungary, July 6-8th, 1992, Elsevier Science Pub., pp.101-119.

82. RUMELHART, D. E. and McCLELLAND, J. L., "Parallel Distributed Processing", vol.1: Foundations, A Bradford Book, 1988.

83. KARRI, V. and FROST, F., "An Intelligent System for Detection of Failed Aluminium Wheels", Proc. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA97), Gold Coast, Australia, Feb. 10-12th, 1997, pp. 147-151.

84. DAYHOFF, J. E., "Neural Network Architectures - An Introduction", Van Nostrand Reinhold, 1990.

85. JONES, C. R. and TSANG, C. P., "On The Convergence of Feed Forward Neural Networks Incorporating Terminal Attractors", The Institute of Electrical and Electronic Engineers, New York, 1996.

86. BRUNELLI, R., "Training Neural Nets Through Stochastic Minimisation", Neural Networks, vol. 7, no. 9, pp. 1405-1412, 1994.

87. HOPFIELD, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. Natl, Acad. Sci., vol. 79, 1982, pp. 2554-2558.

88. TANK, D. W. and HOPFIELD, J. J., "Simple Neural Optimisation Networks: An A/D Converter, Signal Decision Circuit and a Linear Programming Circuit", Institute of Electrical and Electronic Engineers Trans. Circ. Syst., vol. CAS-33, no. 5, 1986, pp. 533-541.

89. ZEIDENBERG, M., "Neural Networks in Artificial Intelligence", New York, Ellis Horwood, 1990.

90. KOHONEN, T., "Associative Memory: A System-Theoretical Approach", Berlin, Springer-Verlag, 1977.

91. KOHONEN, T., "Self-Organisation and Associative Memory", Berlin, Springer-Verlag, 1984.

92. KOHONEN, T., "Adaptive, Associative and Self-Organisation Functions in Neural Computing", Applied Optics, vol. 26, 1987, pp. 4910-4918.

93. KOHONEN, T., "Self-Organised Formation of topologically Correct Feature Maps", Biological Cybernetics, vol. 43, 1982, pp. 59-69.

94. KOHONEN, T., "An Introduction to Neural Computing", Neural Networks, vol. 1, 1988, p. 4.

95. SANTOSO, E. A., "Monitoring of CNC Turning Operations Using Artificial Intelligent Neural Networks", Honours Master of Engineering Science Thesis, University of Wollongong, Dec. 1997.

96. MAREN, A., HARSTON, C. and PAP, R., "Handbook of Neural Network Computing", Academic Press Inc., UK, 1990.

97. NAYLOR, J. and Li, K. P., "Speaker Recognition Using Kohonen's Self-Organising Feature Map Algorithm", First Annual INNS Meeting, Boston, MA, 1988, p. 310.

98. UDO, G. J., "Neural Network Applications in Manufacturing Processes", Computers and Industrial Engineering", vol. 23, no. 1-4, 1992, pp. 97-100.

99. LEE, S. and PARK, J., "Neural Computation for Collision Free Path Planning", Journ. Intelligent Manufacturing, vol. 2, no. 5, 1991, pp. 15-326.

100. GOVEKAR, E. and PEKLENIK, H., "Monitoring of a Drilling Process by Neural Network", 21st CIRP Int. Seminar on Manuf. Systems, Stockholm, Sweden, Jun. 1989.

101. JALEL, N. A., MIRZAI, A. R., LEIGH, J. R. and NICHOLSON, H., "Applications of Neural Networks in Process Control", Neural Network Applications, London, Springer-Verlag, 1991, pp. 101-113.

102. DAVALO, E. and NAIM, P., "Neural Networks", The MacMillan Press Limited, 1991.

103. KARTALOPOULOS, S. V., "Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications", The Institute of Electrical and Electronic Engineers, Inc., New York, 1996.

104. ROSENBLATT, F., "The Perceptron: A Probabilistic Model for Information Storage and Organisation in the Brain", Psychological Review, vol. 65, 1960, pp. 386-408.

105. HORGAN, J., "Can Science Explain Consciousness?", Scientific American, vol. 271, no. 1, 1994, pp. 88-94.

106. VEELENTURF, L. P. J., "Analysis and Applications of Artificial Neural Networks", Prentice Hall International (UK) Limited, 1995.

107. GOLDEN, R. M., Mathematical Methods for Neural Network Analysis and Design", Massachusetts Institute of Technology, 1996.

108. WATSON, M., "Common LISP Modules: Artificial Intelligence in the Era of Neural Networks and Chaos Theory", Springer-Verlag New York, Inc., 1991.

109. FAUSETT, L., "Fundamentals of Neural Networks: Architectures, Algorithms and Applications", Prentice Hall International, Inc., 1994.

110. RITTER, H., MARTINETZ, T. and SCHULTEN, K., "Neural Computation and Self-Organising Maps: An Introduction", Addison-Wesley Publishing Company, 1992.

111. ELSIMARY, H., MASHALI, S. and SHAHEEN, S., "A Method for Training Feed Forward Neural Networks to be Fault Tolerant", Neural Networks Theory, The Institute of Electrical and Electronic Engineers, Inc., New York, 1996.

112. MEHROTRA, K. G., MOHAN, C. K. and RANKA, S., "Bounds on the Number of Samples Needed for Neural Learning", Institute of Electrical and Electronic Engineers Transactions on Neural Networks, vol. 2, no. 6, Nov. 1991, pp. 548-558.

113. CARPENTER, G. A. and GROSSBERG, S., "Neural Networks for Vision and Image Processing", Massachusetts Institute of Technology, 1992.

114. "Conversion of Bauxite Ore to Aluminium Metal: Electrolysis Cell - Hall Process",http://elmhcx9.elmhurst.edu/~chm/onlcourse/chm110/outlines/aluminum. html, 13th Oct. 1998.

115. TABACHNICK, B. G. and FIDELL, L. S., "Using Multivariate Statistics", Second Edition, Harper Collins Publishers, Inc., 1989.

116. LANE, S. H., FLAX, M. G., HANDLEMAN, D. A. and GELFAND, J. J., "Multi-Layer Perceptrons with B-Spline Receptive Field Functions", Advances in Neural Information Processing Systems 3, San Mateo, CA, Morgan Kaufmann, 1991, pp. 684-692.

117. LOWE, D., "What Have Neural Networks To Offer Statistical Pattern Processing?", Proc. SPIE Conference on Adaptive Signal Processing, San Diego, CA, 1991, pp. 460-471.

118. "Radial Basis Function Network", http://nastol.astro.lu.se/~henrik/rbf.html, 13th October 1998.

119. BROOMHEAD, D. S. and LOWE, D., "Multi-Variable Functional Interpolation and Adaptive Networks", Complex Systems 2, 1988, pp. 321-355.

120. MOODY, J. E. and DARKEN, C. J., "Fast Learning in Networks of Locally Tuned Processing Units", Neural Computation 1, 1989, pp. 281-294.

121. RENALS, S., "Radial Basis Function Network for Speech pattern Classification", Electronics Letters 25, 1989, pp. 437-439.

122. POGGIO, T. and GIROSI, F., "Networks for Approximating and Learning", Proc. Institute of Electrical and Electronic Engineers, vol. 78, no. 9, 1990, pp. 1481-1497.

123. SAHA, A., CHRISTIAN, J., TANG, D. S. and Wu, C. L., "Oriented Non-Radial Basis Functions for Image Coding and Analysis", Advances in Neural Information Processing Systems 3, San Mateo, CA, Morgan Kaufmann, 1991, pp. 728-734.

124. POGGIO, T. and EDELMAN, S., "A Network That Learns to Recognise Three Dimensional Objects", Nature (London), vol. 343, 1990, pp. 263-266.

125. NG, K. and LIPPMANN, R. P., "Practical Characteristics of Neural Network and Conventional Pattern Classifiers", Advances in Neural Information Processing Systems 3, San Mateo, CA, Morgan Kaufmann, 1991, pp. 970-976.

126. NIRANJAN, M. and FALLSIDE, F., "Neural Networks and Radial Basis Functions in Classifying Static Speech Patterns", Computer Speech and Language, vol. 4, 1990, pp. 275-289.

127. HE, X. and LAPEDES, A., "Nonlinear Modelling and Prediction by Successive Approximation Using Radial Basis Functions", Technical Report LA-UR-91-1375, Los Alamos National Library, Los Alamos, NM, 1991.

128. KADIRKAMANATHAN, V., NIRANJAN, M. and FALLSIDE, F., "Sequential Adaptation of Radial Basis Function Neural Networks", Advances in Neural Information Processing Systems 3, San Mateo, CA, Morgan Kaufmann, 1991, pp. 721-727.

129. LOWE, D. and WEBB, A. R., "Exploiting Prior Knowledge in Network Optimisation: An Illustration from Medical Prognosis", Network 1, 1990, pp. 299-323.

130. ORR, M. J. L., "Radial Functions", http://www.cns.edu.ac.uk/people/mark/intro/ node7.html, 19th Oct. 1998.

131. PARK, J. and SANDBERG, I. W., "Universal Approximation Using Radial Basis Function Networks", Neural Computation, vol. 3, 1991, pp. 274-275.

132. GIROSI, F. and POGGIO, T., "Networks and the Best Approximation Property", Biological Cybernetics, vol. 63, 1990, pp. 169-176.

133. SONG, X. M., "Radial Basis Function Networks", http://www.cs.helsinki.fi/ ~xianming/thesis/m_conten.html, 13th Oct. 1998.

134. LOWE, D., "Radial Basis Function Networks", Neural Computing Research Group, Aston University, Aston Triangle, Birmingham, 1988, pp. 1-14.

135. TAYLOR, M. P., "Challenges in Optimising and Controlling the Electrolyte in Aluminium Smelters", Proc. International Conference on Molten Slags, Fluxes and Salts, Sydney, Australia, 1997, pp. 659-674.

136. CHEN, C. H., "Fuzzy Logic and Neural Network Handbook", McGraw-Hill Companies, Inc., 1996.

137. CHEN, C. H., "Fuzzy Logic and Neural Network Handbook", The McGraw-Hill Companies, Inc., 1996, pp. 3.1- 3.44.

138. SPECHT, D. F., "General Regression Neural Networks", Institute of Electrical and Electronic Engineers Transactions on Neural Networks, vol. 2, no. 6, Nov. 1991, pp. 568-576.

139. MASTERS, T., "Advanced Algorithms for Neural Networks: A C++ Sourcebook", John Wiley and Sons, 1995.

140. SHAFFER, R., "General Regression Neural Networks", http://cheml.nrl.navy/ ~shatter/grnn.html, 17th Nov. 1998.

141. SARLE, W., "FAQ for comp.ai.neural-net, What is a GRNN?", part 2, ftp://ftp.sas.com/pub/neural/FAQ.html, 1997.

142. KARRI, V, FROST, F, 'An Intelligent System for Detection of Failed Aluminum Wheels', International Conference on Computational Intelligence and Multimedia Applications, Gold Coast, Australia, 107-111 (1997)

143. http://chemdiv-www.nrl.navy.mil/6110/sensors/chemometrics/grnn.html, August 2000.

144. WARREN SARLE , "What is GRNN?", Chemometrics Research Group, FAQ for comp.ai.neural-net, part 2,

145. D.F. SPECHT, A General Regression Neural Network, IEEE Transactions on Neural Networks, 2, 1991, 568-576.

146. A.J.T. SCARR. Metrology and precision engineering. McGraw-Hill Publishing Company Limited, 1967.

147. "Turning", Production Technology, Pg 89-96, Tata McGraw-Hill Publishing Company, 1980.

APPENDIX A

CALIBRATION DATA

- Accelorometer calibration chart
- Dynamometer calibration data

A1. ACCELEROMETER CALIBRATION CHART

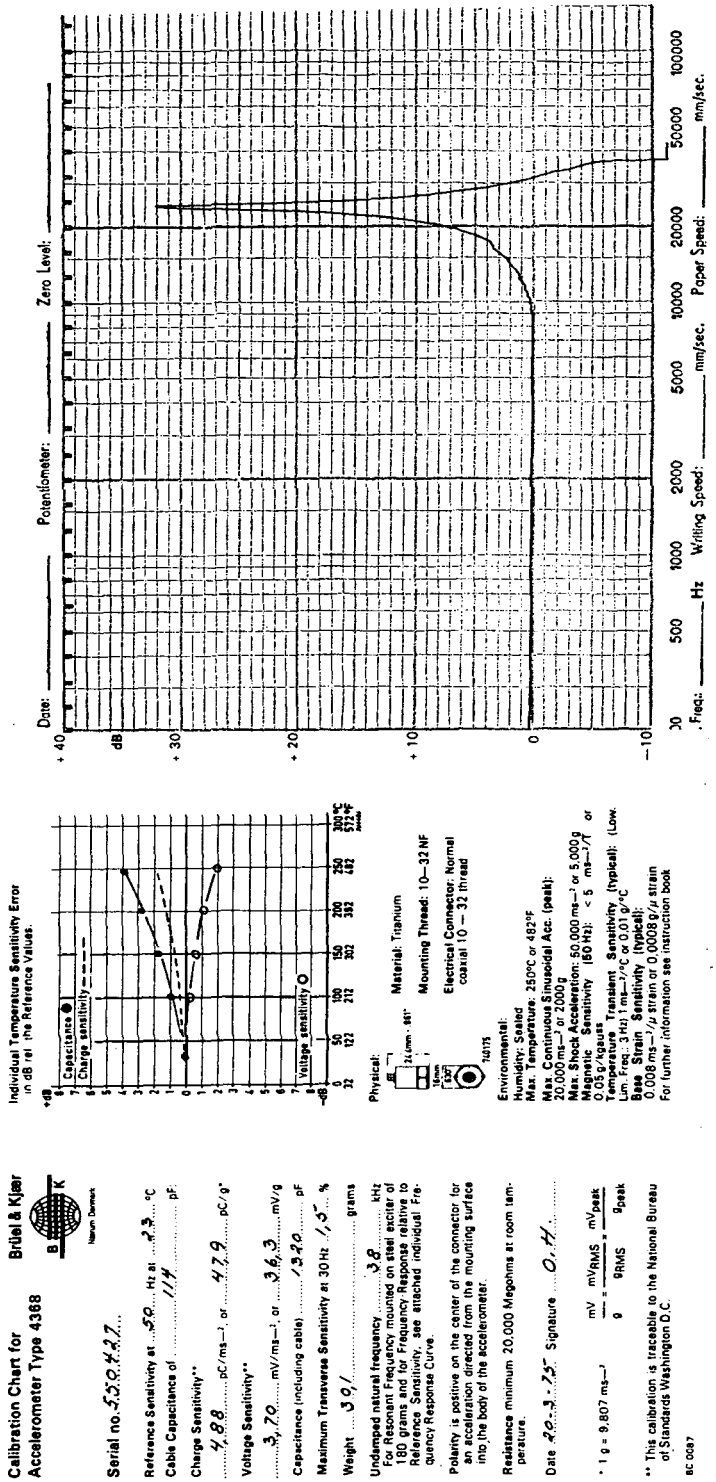

Figure A1: Calibration chart for the 4368 type amplifier.

A2. DYNAMOMETER CALIBRATION DATA

Calibration of dynamometer

Tangential Force		Feed Force	
LOAD (N)	Voltage	LOAD (N)	Voltage
0	-0.113	0	-0.057
10	-0.104	10	-0.096
20	-0.098	20	-0.106
30	-0.096	30	-0.115
40	-0.093	40	-0.124
50	-0.09	50	-0.133
60	-0.087	60	-0.14
70	-0.084	70	-0.145
80	-0.082	80	-0.152
90	-0.08	90	-0.162
100	-0.077	100	-0.167
110	-0.075	110	-0.176
120	-0.071	120	-0.182
150	-0.063	130	-0.207
200	-0.045	140	-0.216
300	-0.022	150	-0.225
400	0.01	200	-0.261
500	0.033	250	-0.294
		300	-0.332
		400	-0.408
		500	-0.486

Table A1: Calibration data for the dynamometer.

APPENDIX B

COMPUTER RELATED PROGRAMS

- Multi output neural network
- Labview data acquisition (reference)

B1 NEURAL NETWORK PROGRAM CODE

```
program Back_Prop_2_Hidden; {2 hidden layers, multiple
inputs/outputs}
uses Dos, crt;
const

{*********************************************************************
*****}

{Neural Network Parameter Specification}

  NumInputs        = 5;
  NumOutputs       = 2;

  TrainPatterns    = 194;
  TestPatterns     = 22;

  InputPatterns    = 22;

  MaxIterations    = 1000;

  NumHiddenNodes   = 5;
  NumHidden2Nodes  = 5;

  linear           = 0;
  sigmoidal        = 1;

  DataDirectory    = 'c:\TimBP\';

  ImportAnal       = 'pi';

  ParamIndex       = '5';

{*********************************************************************
*****}

{Delta Rule constants}

  zeta             = 0.9;  {Controls the learning rate, 0 < zeta < 1}
  decrate          = 0.99; {Rate of decrease of zeta over iterations
range, decrate < 1}

{Control constants}

  EpochSize        = TrainPatterns; {Number of training data patterns
considered in each epoch}

{Miscellaneous constants}

  dataSeed         = 1; {Seed for Random Number Generation for
initialising network weights}
  calcSeed         = 1; {Seed for Random Number Generation for random
selection of training data patterns}
```

```
{File names}

  dataIn         = DataDirectory + ImportAnal + 'trn' + ParamIndex +
'.txt'; {Input file containing training data set}
  testDataIn     = DataDirectory + ImportAnal + 'tst' + ParamIndex +
'.txt'; {Input file containing test data set}
  errorOut       = DataDirectory + ImportAnal + 'err' + ParamIndex +
'.out'; {Output file for training and test error}
  trainingOut    = DataDirectory + ImportAnal + 'trn' + ParamIndex +
'.out'; {Output file for training results}
  testOut        = DataDirectory + ImportAnal + 'tst' + ParamIndex +
'.out'; {Output file for test results}
  weightsOut     = DataDirectory + ImportAnal + 'wts' + ParamIndex +
'.out'; {Output file for weights matrix}
  time           = DataDirectory + ImportAnal + 'tim' + ParamIndex +
'.out'; {Output file for computation time}

  inputDataFile = DataDirectory + 'input.txt';  {Input file
containing validation data set}
  netOutputfile = DataDirectory + 'output.out'; {Output file for
validation results}

type

{Miscellaneous ranges}

  DataRange       = 1..TrainPatterns; {Specifies range of training
data patterns}
  TestDataRange   = 1..TestPatterns;  {Specifies range of test data
patterns}
  InputDataRange  = 1..InputPatterns; {Specifies range of validation
data patterns}
  IterationsRange = 0..MaxIterations; {Specifies range used for
controlling maximum epochs}
  SamplesRange    = 1..EpochSize;     {Specifies range used for
controlling epoch size}

{Network layer ranges}

  InputRange   = 0..NumInputs;       {index i always used for this
range}
  HiddenRange  = 1..NumHiddenNodes;  {index j always used for this
range}
  Hidden2Range = 1..NumHidden2Nodes; {index b always used for this
range}
  OutputRange  = 1..NumOutputs;      {index k always used for this
range}

{Input data types}

  InputLayerType    = array[InputRange] of real;

  OutputOutType     = array[OutputRange] of real;

  InputPEType       = record
                        x: InputLayerType;
                    out: OutputOutType;
                      end;

  InPEPtr           = ^InputPEType;
```

```
                    {Structure too large for stack, put it on the
heap}

  DataType          = array[DataRange] of InPEPtr;

  TestDataType      = array[TestDataRange] of InPEPtr;

  InputDataType     = array[inputDataRange] of InPEPtr;

{Hidden layer 1 data types}

  HiddenWeightType  = array[HiddenRange, InputRange] of real;

  HiddenOutType     = array[HiddenRange] of real;

  HiddenPEType      = record
                        w: HiddenWeightType;
                    z: HiddenOutType;
                      end;

{Hidden layer 2 data types}

  Hidden2WeightType = array[Hidden2Range, HiddenRange] of real;

  Hidden2OutType    = array[Hidden2Range] of real;

  Hidden2PEType     = record
                        r: Hidden2WeightType;
                    s: Hidden2OutType;
                      end;

{Output data types}

  OutputWeightType  = array[OutputRange, Hidden2Range] of real;

  OutputPEType      = record
                        u: OutputWeightType;
                    y: OutputOutType;
                      end;

  DeltaErrorType    = array[OutputRange] of Real;

var

{Network variables}

  inData          : DataType;         {Variable associated with
training data patterns}
  testData        : TestDataType;     {Variable associated with test
data patterns}
  inputData       : InputPEType;      {Variable associated with
validation data patterns}
  inputDataVar    : inputDataType;    {Variable associated with
validation data patterns}
  hiddenNodes     : HiddenPEType;     {Variable used to control
hidden layer 1 computation}
  hidden2Nodes    : Hidden2PEType;    {Variable used to control
hidden layer 2 computation}
  outputNodes     : OutputPEType;     {Variable used to control
output layer computation}
```

```
  delta          : DeltaErrorType;    {Variable used to determine
network error}
  rmsError       : real;              {Variable used to determine
network error}

  iterations     : IterationsRange;   {Variable used to perform
maximum iterations}
  q              : Integer;           {Variable used in random
selection of training patterns}

  fOut           : Text;              {Variable for opening output
files}
  fIn            : Text;              {Variable for opening input
files}

  h1,m1,s1,hund1 : Word;              {Variables for calculating
program start time}
  h2,m2,s2,hund2 : Word;              {Variables for calculating
program finish time}


{***************************** I/O Functions
*****************************}

procedure OpenIn (var f: Text; filename: string);

{Used to open specified files containing information to be read into
the network}

begin {OpenIn}

  Assign(f, filename);
  Reset(f);

end;  {OpenIn}


procedure OpenOut (var f: Text; filename: string);

{Used to open specified files to write network output to}

begin {OpenOut}

  Assign(f, filename);
  Rewrite(f);

end;  {OpenOut}


{************************** Utility Functions
**************************}


function RandomOne: real;

{Returns a random number in the range -1 to 1}

begin {RandomOne}

  RandomOne := Random * 2 - 1;
```

```
end;  {RandomOne}

function Random1To (topOfRange: Integer): DataRange;
{Returns a random integer in the range 1 to topOfRange}
begin {Random1To}
  Random1To := Round(Random * (topOfRange - 1)) + 1;
end;  {Random1To}

function LeadingZero (w : Word) : String;
{Enables computation time to be formatted correctly in output file}
var   s : String;
begin {LeadingZero}
  Str(w:0,s);
  if Length(s) = 1 then
    s:='0'+s;
  LeadingZero:=s;
end;  {LeadingZero}

{********************** Initialization Functions
**********************}

procedure InitDataStrs (var d: DataType; var t: TestDataType);
{Initialises the training and test data structures on the heap}
var  m: DataRange;
     n: TestDataRange;
begin {InitDataStrs}
  for m := 1 to TrainPatterns do
    New(d[m]);
  for n := 1 to TestPatterns do
    New(t[n]);
end;  {InitDataStrs}

procedure DisposeDataStrs (var d: DataType; var t: TestDataType);
{Disposes of the training and test data structures}
```

```
var  m: DataRange;
     n: TestDataRange;

begin {DisposeDataStrs}

  for m := 1 to TrainPatterns do

    Dispose(d[m]);

  for n := 1 to TestPatterns do

    Dispose(t[n]);

end;  {DisposeDataStrs}


procedure InitData (var d: DataType; var t: TestDataType);

{Reads in training and testing data from specified files}

var  m: DataRange;
     n: TestDataRange;
     i: InputRange;
     k: OutputRange;

begin {InitData}

  writeln;
  writeln('Reading in the training data');
  writeln;
  OpenIn(fIn, dataIn);

  for m := 1 to TrainPatterns do

   begin {m}

    for i := 1 to NumInputs do

      read(fIn, d[m]^.x[i]);

    for k := 1 to NumOutputs do

      read(fIn, d[m]^.out[k]);

    readln(fIn);

   end;  {m}

  Close(fIn);

  for m := 1 to TrainPatterns do

    d[m]^.x[0] := 1;

  writeln('Reading in the test data');
  writeln;
  writeln(MaxIterations,' iterations will now commence to generate
network weights');
  writeln;
  OpenIn(fIn, testDataIn);
```

```
  for n := 1 to TestPatterns do

   begin {n}

    for i := 1 to NumInputs do

      read(fIn, t[n]^.x[i]);

    for k := 1 to NumOutputs do

      read(fIn, t[n]^.out[k]);

    readln(fIn);

   end;  {n}

  Close(fIn);

  for n := 1 to TestPatterns do

    t[n]^.x[0] := 1;

end;  {InitData}


procedure InitHidden1Layer (var h: HiddenPEType);

{Initializes the hidden layer 1 weights to random reals in the range
-1 to 1}

var  i: InputRange;
     j: HiddenRange;

begin {InitHidden1Layer}

  for j := 1 to NumHiddenNodes do

    for i := 0 to NumInputs do

      h.w[j, i] := RandomOne;

end;  {InitHidden1Layer}


procedure InitHidden2Layer (var f: Hidden2PEType);

{Initializes the hidden layer 2 weights to random reals in the range
-1 to 1}

var  j: HiddenRange;
     b: Hidden2Range;

begin {InitHidden2Layer}

  for b := 1 to NumHidden2Nodes do

    for j := 1 to NumHiddenNodes do

      f.r[b, j] := RandomOne;

end;  {InitHidden2Layer}
```

```
procedure InitOutputLayer (var o: OutputPEType);

{Initializes the output weights to random reals in the range -1 to
1}

var  b: Hidden2Range;
     k: outputRange;

begin {InitOutputLayer}

  for k := 1 to NumOutputs do

    for b := 1 to NumHidden2Nodes do

      o.u[k, b] := RandomOne;

end;  {InitOutputLayer}


{********************** Neural Net Procedures
**********************}

procedure NetForwardDel (var inp: InputPEType; var h: HiddenPEType;
var f: Hidden2PEType;
var o: OutputPEType; var delta: DeltaErrorType);

{Calculates network output for a given input and also the error,
delta}

var  net, val, sum: real;
     i: InputRange;
     j: HiddenRange;
     b: Hidden2Range;
     k: OutputRange;

begin {NetForwardDel}

{Hidden layer 1 forward}

  for j := 1 to NumHiddenNodes do

  begin {j}

    net := 0;

    for i := 0 to NumInputs do

      net := net + h.w[j, i] * inp.x[i];

    h.z[j] := 1 / (1 + exp(-net));

  end; {j}

{Hidden layer 2 forward}

  for b := 1 to NumHidden2Nodes do

   begin {b}
```

```
    sum := 0;

    for j := 1 to NumHiddenNodes do

      sum := sum + f.r[b,j] * h.z[j];

    f.s[b] := 1 / (1 + exp(-sum));

   end;  {b}

{Output layer forward}

  for k := 1 to NumOutputs do

   begin {k}

    val := 0;

    for b := 1 to NumHidden2Nodes do

      val := val + o.u[k, b] * f.s[b];

    o.y[k] := sigmoidal * (1 / (1 + exp(-val))) + linear * val;
    delta[k] := inp.out[k] - o.y[k];

   end;  {k}

end;  {NetForwardDel}


procedure NetForward (var inp: InputPEType; var h: HiddenPEType; var
f: Hidden2PEType;
var o: OutputPEType; var d1,d2:OutputOutType);

var  net, val, sum: real;
     i: InputRange;
     j: HiddenRange;
     b: Hidden2Range;
     k: OutputRange;

begin {NetForward}

{Hidden layer forward}

  for j := 1 to NumHiddenNodes do

    begin {j}

      net := 0;

      for i := 0 to NumInputs do

        net := net + h.w[j, i] * inp.x[i];

      h.z[j] := 1 / (1 + exp(-net));

    end;  {j}

{Hidden layer 2 forward}

  for b := 1 to NumHidden2Nodes do
```

```
   begin {b}
    sum := 0;
    for j := 1 to NumHiddenNodes do
      sum := sum + f.r[b,j] * h.z[j];
    f.s[b] := 1 / (1 + exp(-sum));
   end;  {b}
{Output layer forward}
  for k := 1 to NumOutputs do
   begin {k}
    val := 0;
    for b := 1 to NumHidden2Nodes do
      val := val + o.u[k, b] * f.s[b];
    o.y[k] := sigmoidal * (1 / (1 + exp(-val))) + linear * val;
    d1[k] := inp.out[k];
    d2[k] := o.y[k];
   end;  {k}
end;  {NetForward}


procedure NetTrain (var inp: InputPEType; var h: HiddenPEType; var
f: Hidden2PEType;
var o: OutputPEType; newzeta : real);

{This procedure calculates an output for any given input, compares
the predicted output
 with the given output, calculates the associated error, delta, and
updates the network
 weights using the "delta rule"}

var  i: InputRange;
     j: HiddenRange;
     k: OutputRange;
     b: Hidden2Range;
     net, val, deltaJ, sum, deltaB, deltaB1, val1: real;

begin {NetTrain}

{Hidden layer forward}

  for j := 1 to NumHiddenNodes do
   begin {j}
    net := 0;
    for i := 0 to NumInputs do
```

```
      net := net + h.w[j, i] * inp.x[i];
    h.z[j] := 1 / (1 + exp(-net));
   end;  {j}
{Hidden layer 2 forward}
  for b := 1 to NumHidden2Nodes do
   begin {b}
    sum := 0;
    for j := 1 to NumHiddenNodes do
      sum := sum + f.r[b,j] * h.z[j];
    f.s[b] := 1 / (1 + exp(-sum));
   end;  {b}
{Output layer forward}
  for k := 1 to NumOutputs do
   begin {k}
    val := 0;
    for b := 1 to NumHidden2Nodes do
      val := val + o.u[k,b] * f.s[b];
    o.y[k] := sigmoidal * (1 / (1 + exp(-val))) + linear * val;
    delta[k] := inp.out[k] - o.y[k];
   end;  {k}
{Update output layer weights}
  for k := 1 to NumOutputs do
    for b := 1 to NumHidden2Nodes do
      o.u[k, b] := o.u[k, b] + newzeta * delta[k] * f.s[b];
{Update hidden layer 2 weights}
  for b := 1 to NumHidden2Nodes do
   begin {b}
    val := 0;
    for k := 1 to NumOutputs do
     val := val + delta[k] * o.u[k, b];
    deltaB := f.s[b] * (1 - f.s[b]) * val;
```

```
    for j := 1 to NumHiddenNodes do

     f.r[b, j] := f.r[b, j] + newzeta * deltaB * h.z[j];

   end;  {b}

{Update hidden layer 1 weights}

   for j := 1 to NumHiddenNodes do

    begin {j}

     sum := 0;
     val1 := 0;

     for b := 1 to numHidden2nodes do

      begin {b}

        for k := 1 to NumOutputs do

          val1 := val1 + delta[k] * o.u[k, b];

        deltaB1 := f.s[b] * (1 - f.s[b]) * val1;
        sum := sum + deltaB1 * f.r[b,j];
        deltaJ := h.z[j] * (1 - h.z[j]) * sum;

        for i := 0 to NumInputs do

          h.w[j, i] := h.w[j, i] + zeta * deltaJ * inp.x[i];

      end;  {b}

    end;  {j}

end;  {NetTrain}


{************************ Display Procedures
************************}


procedure DisplayError (var rms: real; inD: DataType; tD:
TestDataType; h: HiddenPEType;
f: Hidden2PEType; o: OutputPEType; newzeta: real);

{Calculates the Root Mean Square, RMS, error for all the training
and test inputs}
{Displayed on user screen for monitoring purposes and also written
to the specified file}

var  trErr, testErr, val, testRms: real;
     del: DeltaErrorType;
     m: DataRange;
     n: TestDataRange;
     k: OutputRange;

begin {DisplayError}

{Training data set calculations}
```

```
  trErr := 0;

  for m := 1 to TrainPatterns do

   begin {m}

    NetForwardDel(inD[m]^, h, f, o, del);
    val := 0;

    for k := 1 to NumOutputs do

      val := val + del[k] * del[k];

    trErr := trErr + val;

   end;  {m}

   rms := Sqrt(trErr/TrainPatterns); {RMS error over all training
patterns}

{Test data set calculations}

  testErr := 0;

  for n := 1 to TestPatterns do

   begin {n}

    NetForwardDel(tD[n]^, h, f, o, del);
    val := 0;

    for k := 1 to NumOutputs do

      val := val + del[k] * del[k];

    testErr := testErr + val;

   end;  {n}

   testRms := Sqrt(testErr/TestPatterns); {RMS error over all test
patterns}

  writeln('It. ', iterations : 3, '  e1= ', rms : 5 : 5, '  e2= ',
testRms : 5 : 5, ' ', 'zeta = ', newzeta : 3 : 2);
  writeln(fOut, iterations : 3, '  ', rms : 5 : 5, '  ', testRms : 5
: 5);

end;  {DisplayError}


procedure WriteOutTrainingData (var inD: DataType; var h:
HiddenPEType; var f: Hidden2PEType;
var o: OutputPEType);

{Writes the predicted and actual outputs from training to the
specified file}

var  curve1, curve2: OutputOutType;
     p: DataRange;
     k: OutputRange;
```

```
begin {WriteOutTrainingData}

  OpenOut(fOut, trainingOut);

  for p := 1 to TrainPatterns do

   begin {p}

    NetForward(inD[p]^, h, f, o, curve1, curve2);

    for k :=1 to NumOutputs do

      write(fOut, curve1[k] : 7 : 5, '  ');

    for k :=1 to NumOutputs do

      write(fOut, curve2[k] : 7 : 5, '  ');

    writeln(fOut);

   end;  {p}

  Close(fOut);

end;  {WriteOutTrainingData}


procedure WriteOutTestData (var tD: TestDataType; var h:
HiddenPEType; var f: Hidden2PEType;
var o: OutputPEType);

{Writes the predicted and actual outputs from testing to the
specified file}

var  curve1, curve2: OutputOutType;
     q: TestDataRange;
     k: OutputRange;

begin {WriteOutTestData}

  OpenOut(fOut, testOut);

  for q := 1 to TestPatterns do

   begin {q}

    NetForward(tD[q]^, h, f, o, curve1, curve2);

    for k := 1 to NumOutputs do

      write(fOut, curve1[k] : 7 : 5, '  ');

    for k := 1 to NumOutputs do

      write(fOut, curve2[k] : 7 : 5, '  ');

    writeln(fOut);

   end;  {q}
```

```
  Close(fOut);

end;  {WriteOutTestData}


procedure WriteOutWeights (var h: HiddenPEType; var f:
Hidden2PEType; var o: OutputPEType);

{Writes the weights matrix to the specified file}

var  i: InputRange;
     j: HiddenRange;
     b: Hidden2Range;
     k: OutputRange;

begin {WriteOutWeights}

  OpenOut(fOut, weightsOut);

  for j := 1 to NumHiddenNodes do

   begin {j}

    for i := 0 to NumInputs do

      write(fOut, h.w[j, i] : 8 : 5, '  ');

    writeln(fOut);

   end;  {j}

  for b := 1 to NumHidden2Nodes do

   begin {b}

    for j := 1 to NumHiddenNodes do

     write(fOut, f.r[b,j] : 8 : 5,' ');

    writeln(fOut);

   end;  {b}

  writeln(fOut);

  for j := 1 to NumHiddenNodes do

   begin {j}

    for k := 1 to NumOutputs do

      Write(fOut, o.u[k,j] : 8 : 5, '  ');

    writeln(fOut);

   end;  {j}

  Close(fOut);

end;  {WriteOutWeights}
```

```
{Procedures from here to "main program" used for running option 2 of
the program}


procedure InitInputData (var d: inputDataType);

{Reads validation data patterns from specified file}

var  m: inputDataRange;
     i: InputRange;
     k: OutputRange;

begin {InitInputData}

  writeln('Reading in the input data');
  OpenIn(fIn, inputDataFile);

  for m := 1 to InputPatterns do

   begin {m}

    for i := 1 to NumInputs do

     read(fIn, d[m]^.x[i]);

    for k := 1 to NumOutputs do

     read(fIn,d[m]^.out[k]);

   end;  {m}

    for m := 1 to InputPatterns do

     d[m]^.x[0] := 1; {}

  Close(fIn);

end;  {InitInputData}

procedure ReadInWeights(var h:HiddenPEType; var f:Hidden2PEType; var
o:OutputPEType);

{Reads weights matrix from file produced using option 1 of the
program}

var  i, j, b, k : integer;

begin {ReadInWeights}

  OpenIn(fIn,weightsOut);

  for j := 1 to NumHiddenNodes do

   begin {j}

    for i := 0 to NumInputs do

     read(fIn, h.w[j, i]) ;
     readln(fIn);
```

```
    end;  {j}

    for b := 1 to NumHidden2Nodes do

     begin {b}

      for j := 1 to NumHiddenNodes do

       read(fIn, f.r[b, j]);
       readln(fIn);

     end;  {b}

    readln(fIn);

     for j := 1 to NumHiddenNodes do

      begin {j}

       for k := 1 to NumOutputs do

        read(fIn, o.u[k,j]);
        readln(fIn);

      end;  {j}

  Close(fIn);

end;  {ReadInWeights}

procedure WriteOutNetOutputData (var tD: inputDataType; var h:
HiddenPEType; var f:Hidden2PEType; var o: OutputPEType);

{Calculates the output for any given input using the weights
developed during training}

var  curve1, curve2: OutputOutType;
     q: inputDataRange;
     k:OutputRange;

begin {WriteOutTestData}

  OpenOut(fOut, netOutputFile);

  for q := 1 to InputPatterns do

   begin {q}

    NetForward(tD[q]^, h, f, o, curve1, curve2);

    for k := 1 to NumOutputs do

     write(fOut, curve1[k] : 7 : 5, '  ');

    for k := 1 to NumOutputs do

     write(fOut, curve2[k] : 7 : 5, '  ');
     writeln(fOut);

   end;  {q}
```

```
  Close(fOut);

end;  {WriteOutTestData}

procedure InitInputDataStrs (var d: inputDataType);

{Initialises the training and test data structures on the heap}

var  m: inputDataRange;

begin {InitDataStrs}

  for m := 1 to InputPatterns do

   New(d[m]);

end;  {InitDataStrs}


{*************************  Main Program
*************************}

var  answer: integer;
     newzeta : real;

begin {Main program}

 clrscr;

 writeln;
 writeln;
 writeln('  FeedForward BackPropagation Neural Network');
 writeln('  ----------------------------------------');
 writeln;
 writeln;
 writeln('  Please Select Item (1 or 2) Then Press ENTER');
 writeln;
 writeln;
 writeln('  1. Train and Test the Network');
 writeln;
 writeln('     (Uses training and test data patterns to generate
network weights)');
 writeln;
 writeln;
 writeln('  2. Use the Network');
 writeln;
 writeln('     (Uses generated network weights to create output for
new data)');
 writeln;
 read(answer);

 if (answer = 2) then

  begin {2}

    InitinputDataStrs (inputDataVar);
    ReadInWeights (HiddenNodes, Hidden2Nodes, OutputNodes);
    InitInputData (inputDataVar);
    WriteOutNetOutputData (inputDataVar, HiddenNodes, Hidden2Nodes,
OutputNodes);
```

```
  end   {2}

 else

  begin {1}

    GetTime (h1, m1, s1, hund1);

    {Initialize the training and test sets}

    InitDataStrs (inData, testData);
    InitData (inData, testData);
    OpenOut (fOut, errorOut);

    {Initialise all weights using random values}

    randSeed := dataSeed;
    InitHidden1Layer (hiddenNodes);
    InitHidden2Layer (hidden2Nodes);
    InitOutputLayer (outputNodes);

    {A 'for' loop is used to fix the total number of iterations}

    {Reset the random seed so that the calling sequence can be
contolled}

    randSeed := calcSeed;
    newzeta := zeta;

    for iterations := 1 to MaxIterations do

     begin {iterations}

      for q := 1 to EpochSize do

       begin {q}

        inputData := InData[Random1To(TrainPatterns)]^;
        NetTrain (inputData, hiddenNodes, hidden2Nodes, outputNodes,
newzeta);

       end;  {q}

      {Display RMS error for each iteration}

      DisplayError (rmsError, inData, testData, hiddenNodes,
hidden2Nodes, outputNodes, newzeta);

      {Decrease delta rule constant over iterations range}

      newzeta := decrate * newzeta;

      if newzeta < 0.1 then

       newzeta := 0.1

      else

       newzeta := newzeta;

     end;  {iterations}
```

```
    Close (fOut);
    WriteOutTrainingData (inData, hiddenNodes, hidden2Nodes,
outputNodes);
    WriteOutTestData (testData, hiddenNodes, hidden2Nodes,
outputNodes);
    WriteOutWeights (hiddenNodes, hidden2Nodes, outputNodes);
    DisposeDataStrs (inData, testData);
    GetTime (h2, m2, s2, hund2);
    OpenOut (fOut, time);
    writeln (fOut, 'Start time  :-
',LeadingZero(h1),':',LeadingZero(m1),':',LeadingZero(s1),':',Leadin
gZero(hund1));
    writeln (fOut, 'Finish time :-
',LeadingZero(h2),':',LeadingZero(m2),':',LeadingZero(s2),':',Leadin
gZero(hund2));
    Close(fOut);

  end; {1}

end.  {Main program}
```

B2 DATA ACQUISITION PROGRAM

The data acquisition was obtained through a G code based labview file. Labview and its associated G-code are discussed in depth in chapter 5. This functioning file can be viewed in full on the attached CD, along with diagrams of the front and back panel.

APPENDIX C

EXPERIMENTAL OUTPUT DATA

- Original experimental output
- Test samples (reference)

C1 EXPERIMENTAL OUTPUT DATA

TEST	FEED	SPEED	DEPTH	Tool	Volt zero	Ftang	Volts	FTang	Volts	Ftang	Volts
1	0.03	300	0.5	PF	-0.01	5.555556	-0.005	3.333333	-0.007	3.333333	-0.007
2	0.04	300	0.5		-0.012	5.555556	-0.007	3.333333	-0.009	6.666667	-0.006
3	0.05	300	0.5		-0.015	8.222104	-0.005	8.222104	-0.005	8.222104	-0.005
4	0.06	300	0.5		-0.023	22.39143	-0.009	18.8491	-0.01	11.76443	-0.012
5	0.08	300	0.5		-0.027	36.56075	-0.009	29.47609	-0.011	18.8491	-0.014
6	0.1	300	0.5		-0.036	33.01842	-0.019	29.47609	-0.02	25.93376	-0.021
7	0.12	300	0.5		-0.041	43.64541	-0.021	43.64541	-0.021	40.10308	-0.022
8	0.16	300	0.5		-0.046	64.8994	-0.02	64.8994	-0.02	64.8994	-0.02
9	0.2	300	0.5		-0.018	135.746	0.028	132.2037	0.027	132.2037	0.027
10	0.25	300	0.5		-0.037	231.3889	0.036	224.3043	0.034	224.3043	0.034
11	0.3	300	0.5								
12	0.08	300	0.1	PF	-0.001	2.222222	0.001	3.333333	0.002	0	-0.001
13	0.08	300	0.2		-0.018	5.555556	-0.013	2.222222	-0.016	2.222222	-0.016
14	0.08	300	0.3		-0.029	11.76443	-0.018	7.777778	-0.022	5.555556	-0.024
15	0.08	300	0.4		-0.039	15.30677	-0.027	15.30677	-0.027	8.222104	-0.029
16	0.08	300	0.5		-0.047	43.64541	-0.027	36.56075	-0.029	29.47609	-0.031
17	0.08	300	0.6		-0.058	50.73007	-0.036	43.64541	-0.038	40.10308	-0.039
18	0.08	300	0.7		-0.085	61.35707	-0.06	64.8994	-0.059	57.81474	-0.061
19	0.08	300	0.8		-0.002	75.52639	0.027	75.52639	0.027	71.98406	0.026
20	0.08	300	0.9		-0.009	82.61105	0.022	86.15338	0.023	82.61105	0.022
21	0.08	300	1		-0.018	121.5767	0.024	118.0344	0.023	118.0344	0.023
22	0.08	300	1.1		-0.049	139.2883	-0.002	142.8307	-0.001	135.746	-0.003
23	0.08	70	0.5	PF	-0.037	47.18774	-0.016	47.18774	-0.016	50.73007	-0.015
24	0.08	160	0.5		-0.035	36.56075	-0.017	36.56075	-0.017	36.56075	-0.017
25	0.08	250	0.5		-0.037	25.93376	-0.022	25.93376	-0.022	33.01842	-0.02
26	0.08	340	0.5		-0.038	18.8491	-0.025	22.39143	-0.024	22.39143	-0.024
27	0.08	430	0.5		-0.036	15.30677	-0.024	15.30677	-0.024	15.30677	-0.024
28	0.08	520	0.5		-0.036	15.30677	-0.024	15.30677	-0.024	15.30677	-0.024
29	0.08	610	0.5		-0.037	15.30677	-0.025	11.76443	-0.026	11.76443	-0.026
30	0.08	700	0.5		-0.037	11.76443	-0.026	11.76443	-0.026	11.76443	-0.026
31	0.08	790	0.5		-0.037	8.222104	-0.027	8.222104	-0.027	11.76443	-0.026
32	0.08	880	0.5		-0.039	11.76443	-0.028	11.76443	-0.028	11.76443	-0.028
33	0.08	970	0.5		-0.041	22.39143	-0.027	29.47609	-0.025	29.47609	-0.025
34	0.08	1000	0.5		-0.077	29.47609	-0.061	25.93376	-0.062	22.39143	-0.063
35	0.08	1100	0.5		-0.078	36.56075	-0.06	33.01842	-0.061	33.01842	-0.061
36	0.08	1200	0.5		-0.08	40.10308	-0.061	43.64541	-0.06	40.10308	-0.061
37	0.08	1300	0.5		-0.077	29.47609	-0.061	29.47609	-0.061	29.47609	-0.061
38	0.03	300	0.5	PM	-0.071	4.444444	-0.067	3.333333	-0.068	4.444444	-0.067
39	0.04	300	0.5		-0.072	8.888889	-0.064	8.888889	-0.064	8.888889	-0.064
40	0.05	300	0.5		-0.073	8.888889	-0.065	10	-0.064	10	-0.064
41	0.06	300	0.5		-0.078	18.8491	-0.065	18.8491	-0.065	18.8491	-0.065
42	0.08	300	0.5		-0.083	36.56075	-0.065	40.10308	-0.064	29.47609	-0.067
43	0.1	300	0.5		-0.085	40.10308	-0.066	40.10308	-0.066	40.10308	-0.066
44	0.12	300	0.5		-0.055	33.01842	-0.038	29.47609	-0.039	22.39143	-0.041
45	0.16	300	0.5		-0.059	61.35707	-0.034	61.35707	-0.034	57.81474	-0.035
46	0.2	300	0.5		-0.064	96.78038	-0.029	100.3227	-0.028	96.78038	-0.029
47	0.25	300	0.5		-0.066	142.8307	-0.018	142.8307	-0.018	142.8307	-0.018
48	0.3	300	0.5								
49	0.08	300	0.1	PM	-0.08	5.555556	-0.075	6.666667	-0.074	5.555556	-0.075
50	0.08	300	0.2		-0.082	7.777778	-0.075	7.777778	-0.075	7.777778	-0.075
51	0.08	300	0.3		-0.084	15.30677	-0.072	15.30677	-0.072	11.76443	-0.073
52	0.08	300	0.4		-0.092	22.39143	-0.078	18.8491	-0.079	15.30677	-0.08
53	0.08	300	0.5		-0.096	22.39143	-0.082	22.39143	-0.082	25.93376	-0.081
54	0.08	300	0.6		-0.102	36.56075	-0.084	29.47609	-0.086	22.39143	-0.088
55	0.08	300	0.7		-0.108	40.10308	-0.089	29.47609	-0.092	36.56075	-0.09
56	0.08	300	0.8		-0.116	57.81474	-0.092	64.8994	-0.09	57.81474	-0.092
57	0.08	300	0.9		-0.123	68.44173	-0.096	64.8994	-0.097	61.35707	-0.098
58	0.08	300	1		-0.129	82.61105	-0.098	82.61105	-0.098	79.06872	-0.099
59	0.08	300	1.1		-0.136	89.69571	-0.103	89.69571	-0.103	89.69571	-0.103
60	0.08	70	0.5								

Table C.1: Effect of speed, feed, depth of cut and tool on tangential force

61	0.08	160	0.5	-0.02	64.8994	0.006	57.81474	0.004	50.73007	0.002
62	0.08	250	0.5	-0.03	54.27241	-0.007	50.73007	-0.008	36.56075	-0.012
63	0.08	340	0.5	-0.038	54.27241	-0.015	47.18774	-0.017	40.10308	-0.019
64	0.08	430	0.5	-0.05	54.27241	-0.027	47.18774	-0.029	43.64541	-0.03
65	0.08	520	0.5	-0.059	61.35707	-0.034	57.81474	-0.035	50.73007	-0.037
66	0.08	610	0.5	-0.067	61.35707	-0.042	57.81474	-0.043	57.81474	-0.043
67	0.08	700	0.5	-0.037	64.8994	-0.011	64.8994	-0.011	61.35707	-0.012
68	0.08	790	0.5	-0.043	43.64541	-0.023	40.10308	-0.024	36.56075	-0.025
69	0.08	880	0.5	-0.046	50.73007	-0.024	47.18774	-0.025	47.18774	-0.025
70	0.08	970	0.5	-0.053	40.10308	-0.034	36.56075	-0.035	33.01842	-0.036
71	0.08	1000	0.5	-0.057	43.64541	-0.037	40.10308	-0.038	36.56075	-0.039
72	0.08	1100	0.5	-0.064	54.27241	-0.041	47.18774	-0.043	47.18774	-0.043
73	0.08	1200	0.5	-0.068	54.27241	-0.045	50.73007	-0.046	50.73007	-0.046
74	0.08	1300	0.5	-0.074	61.35707	-0.049	61.35707	-0.049	61.35707	-0.049
75	0.08	520	0.5	-0.091	57.81474	-0.067	57.81474	-0.067	57.81474	-0.067
76	0.08	700	0.5	-0.091	64.8994	-0.065	64.8994	-0.065	64.8994	-0.065
77	0.08	880	0.5	-0.088	50.73007	-0.066	50.73007	-0.066	50.73007	-0.066
78	0.08	1060	0.5	-0.089	50.73007	-0.067	50.73007	-0.067	50.73007	-0.067
79	0.08	1200	0.5	-0.091	54.27241	-0.068	54.27241	-0.068	54.27241	-0.068

Table C.2: Effect of speed, feed, depth of cut and tool on tangential force

Volt Zero	Ffeed	Volts	N	Volts	N	Volts	Freq1	Freq2	Freq3	Ra	Ry
-0.008	4.375	-0.015	2.5	-0.012	2.5	-0.012	270	309.4	302.4	2.54	17.8
0.003	7.5	-0.009	7.5	-0.009	3.75	-0.003	310.6	350.1	254.1	2.7	19.8
0.009	6.875	-0.002	6.25	-0.001	6.875	-0.002	310.6	275.8	320	2.56	17.3
0.016	8.75	0.002	9.375	0.001	7.5	0.004	331.4	250.9	335.2	2.72	17
0.015	8.75	0.001	8.75	0.001	9.375	0	288.6	315.2	320.8	3.13	20.6
0.023	12.5	0.003	11.25	0.005	10.625	0.006	292	369.4	320.1	2.99	18.9
0.025	13.75	0.003	13.125	0.004	13.75	0.003	335.6	335.9	279.8	3.32	23.2
0.027	16.25	0.001	15.625	0.002	16.25	0.001	328.3	321.1	320.5	3.47	19.4
-0.009	21.25	-0.043	21.875	-0.044	21.875	-0.044	361.7	361.6	356.2	3.67	22.6
-0.0068	26.375	-0.049	27.625	-0.051	27	-0.05	357.2	377.9	360.7	3.88	21.9
0.01	2.5	0.006	1.25	0.008	0	0.01	166.3	164.8	173.2	3.19	20.4
0.02	3.75	0.014	3.125	0.015	3.75	0.014	275.2	250.6	245.4	3.33	21.2
0.017	6.25	0.007	4.375	0.01	6.25	0.007	244	268.8	224.4	3.03	19.4
0.023	6.875	0.012	6.25	0.013	6.25	0.013	248.2	244.8	270.2	3.06	19.6
0.029	13.75	0.007	13.125	0.008	14.375	0.006	270.9	252.1	253.6	3.16	22.2
0.033	16.875	0.006	16.25	0.007	16.25	0.007	272.5	214	352	3.38	25.6
0.057	19.375	0.026	19.375	0.026	19.375	0.026	348.2	241.2	332.1	3.45	26.3
-0.013	26.25	-0.055	23.75	-0.051	25	-0.053	379	354.2	370.7	3.75	28.6
-0.004	31.875	-0.055	33.125	-0.057	33.125	-0.057	365	336.7	315.3	3.88	25
-0.004	39.375	-0.067	41.25	-0.07	42.5	-0.072	375	404.5	374.5	3.78	26.1
0.04	47.5	-0.036	50.625	-0.041	51.25	-0.042	409.8	365	360.1	3.84	27
-0.006	23.75	-0.044	23.75	-0.044	23.75	-0.044	350.4	374.9	330.6	4.2	38.7
-0.004	17.5	-0.032	16.875	-0.031	18.125	-0.033	342	360	363.2	3.41	28.8
-0.004	15	-0.028	15.625	-0.029	15.625	-0.029	344.2	263.3	249.8	3.7	22.6
-0.004	13.125	-0.025	13.75	-0.026	13.125	-0.025	356.9	364.2	240.2	3.76	33.1
-0.002	14.375	-0.025	15	-0.026	15	-0.026	352.7	317.2	320.8	2.58	15
-0.002	14.375	-0.025	15.625	-0.027	15	-0.026	204.1	243.9	260.7	3.09	20.8
-0.002	12.5	-0.022	12.5	-0.022	12.5	-0.022	231.4	226.1	253.6	2.43	15.6
-0.003	13.125	-0.024	11.875	-0.022	13.75	-0.025	238.2	279.7	228.4	2.71	17.3
-0.003	9.375	-0.018	8.75	-0.017	10	-0.019	299.9	200.2	200.1	2.52	15.8
-0.003	9.375	-0.018	9.375	-0.018	10.625	-0.02	256.4	305	240	2.78	16.8
-0.005	18.75	-0.035	18.75	-0.035	18.125	-0.034	265.5	266.9	229.8	2.23	14
-0.01277	20.28125	-0.04522	20.74375	-0.04596	20.28125	-0.04522	302.9	261.9	363.4	2.46	13.8
-0.01153	21.3875	-0.04575	21.9375	-0.04663	21.3875	-0.04575	272.3	300.3	310.3	2.64	16.5
-0.00967	17.88125	-0.03828	18.01875	-0.0385	17.88125	-0.03828	281.7	281.4	345	1.57	11.2
-0.00497	20.85625	-0.03834	20.89375	-0.0384	20.85625	-0.03834	245.5	210.8	290.4	1.82	9.9
0.014	5.625	0.005	5	0.006	5.625	0.005	274.1	316.8	324.8	2.66	16.3
0.013	5.625	0.004	5.625	0.004	5.625	0.004	220.5	310.7	260.1	2.76	17.4
0.013	6.25	0.003	7.5	0.001	7.5	0.001	304	249.9	280.5	2.85	18.3
0.014	7.5	0.002	7.5	0.002	7.5	0.002	295.4	330.6	320.5	3.07	20.2
0.014	15.625	-0.011	15	-0.01	15	-0.01	357.3	384.6	310.2	3.01	19.9
0.012	18.75	-0.018	18.75	-0.018	18.75	-0.018	342	367.1	345.9	3.33	24.8
0.011	20.625	-0.022	20	-0.021	19.375	-0.02	360.6	369.8	385.7	3.42	27
0.011	21.875	-0.024	22.5	-0.025	23.75	-0.027	267.8	326.3	357.6	3.23	23.5
0.013	26.875	-0.03	26.875	-0.03	26.875	-0.03	378	328.4	381	3.44	28.4
0.012	31.25	-0.038	32.5	-0.04	31.875	-0.039	336.3	339.3	312	3.644	27.7
	0		0		0						
0.015	3.125	0.01	3.125	0.01	3.125	0.01	359.7	262	282.3	3.06	19.6
0.015	3.75	0.009	3.75	0.009	3.125	0.01	345.6	366.5	288.8	3.39	27.3
0.017	10	0.001	9.375	0.002	10.625	0	259.9	346.9	316.5	3.34	29
0.019	13.125	-0.002	11.25	0.001	10	0.003	304	297.8	293.9	3.35	24.2
0.021	15	-0.003	13.125	0	13.75	-0.001	368.1	297.2	252.2	3.42	23
0.019	25.625	-0.022	25.625	-0.022	26.25	-0.023	368	376.1	377.7	3.46	20.9
0.03	28.75	-0.016	30	-0.018	29.375	-0.017	374.7	377.6	357.9	3.54	20
0.027	30.625	-0.022	31.25	-0.023	31.25	-0.023	378.6	379.4	379.7	3.59	22.9
0.031	36.25	-0.027	37.5	-0.029	38.125	-0.03	380.6	374.6	350.3	3.67	21.9
0.031	39.375	-0.032	39.375	-0.032	39.375	-0.032	381.4	381.5	379.4	3.7	29.7
0.035	41.875	-0.032	43.125	-0.034	43.125	-0.034	376.8	370.6	368.6	3.85	34.4
	0		0		0						

Table C.3: Effect of speed, feed, depth of cut and tool on feed force, vibrations and surface finish

0.038	19.375	0.007	18.125	0.009	17.5	0.01	370.6	365.7	378.4	4.67	32.8
0.047	18.75	0.017	18.75	0.017	18.75	0.017	359.5	349.6	338.3	3.85	27.6
0.048	19.375	0.017	19.375	0.017	18.75	0.018	398.2	408.7	360.7	3.73	25.9
0.055	21.25	0.021	21.875	0.02	21.875	0.02	366.2	265.5	405.5	4.38	25.8
0.057	23.75	0.019	23.75	0.019	23.75	0.019	452.9	459.3	392.1	3.54	22.4
0.058	23.125	0.021	23.125	0.021	23.125	0.021	485.7	496.6		3.28	25.2
-0.01	19.375	-0.041	19.375	-0.041	20	-0.042	154.4	431.3	468.4	3.33	21.3
-0.006	22.5	-0.042	23.125	-0.043	22.5	-0.042	390.1	335.6	406.3	3.13	21.1
-0.008	24.375	-0.047	23.75	-0.046	24.375	-0.047	499	106.6	264.7	2.2	19.5
-0.002	20	-0.034	20	-0.034	20	-0.034	376.2	367.4	376.7	1.94	13.6
-0.002	21.875	-0.037	21.25	-0.036	21.875	-0.037	362.3	344.9	361.3	1.54	14.9
-0.001	22.5	-0.037	22.5	-0.037	23.125	-0.038	373.3	179.8	373.3	1.47	9.7
-0.00045	21.6875	-0.03515	21.51875	-0.03488	21.66688	-0.035117	317.5	336.1	336.1	1.32	10.3
0.00009	21.76875	-0.03474	21.76875	-0.03474	21.76875	-0.03474	296.3	296.3	296.3	1.22	11.7
-0.0001	6.5125	-0.01052	6.5125	-0.01052	6.5125	-0.01052	330.8	330.8	330.8	3.54	22.4
-0.00055	7.51875	-0.01258	7.51875	-0.01258	7.51875	-0.01258	427.6	427.6	427.6	3.93	28.3
-0.00044	9.53125	-0.01569	9.53125	-0.01569	9.53125	-0.01569	360.7	360.7	360.7	2.2	19.5
-0.00022	25.6125	-0.0412	25.6125	-0.0412	25.6125	-0.0412	325.2	325.2	325.2	1.14	7.9
-0.00022	8.54375	-0.01389	8.54375	-0.01389	8.54375	-0.01389	499.1	499.1	499.1	1.32	10.3

Table C.4: Effect of speed, feed, depth of cut and tool on feed force, vibrations and surface finish

Av volt tan	Av vold feed	Av N Tan	Av N Feed	Av Freq	V-zero Tan	V-zero feed	Actual Feed	Actual Tang
-0.0063333	-0.013	4.0740741	3.125	293.93333	0.00366667	-0.005	3.125	4.07407407
-0.0073333	-0.007	5.1851852	6.25	304.93333	0.00466667	-0.01	6.25	5.18518519
-0.005	-0.0016667	8.2221041	6.6666667	302.13333	0.01	-0.01066667	6.66666667	8.22210414
-0.0103333	0.00233333	17.66832	8.5416667	305.83333	0.01266667	-0.01366667	8.54166667	17.6683198
-0.0113333	0.00066667	28.295312	8.9583333	308.2	0.01566667	-0.01433333	8.95833333	28.2953123
-0.02	0.00466667	29.476089	11.458333	327.16667	0.016	-0.01833333	11.4583333	29.4760893
-0.0213333	0.00333333	42.464636	13.541667	317.1	0.01966667	-0.02166667	13.5416667	42.4646357
-0.02	0.00133333	64.899398	16.041667	323.3	0.026	-0.02566667	16.0416667	64.8993978
0.02733333	-0.0436667	133.38446	21.666667	359.83333	0.04533333	-0.03466667	21.6666667	133.384461
0.03466667	-0.05	226.66584 0	25.083333	365.26667	0.07166667	-0.0432	27	226.66584
0.00066667	0.008	1.8518519	1.25	168.1	0.00166667	-0.002	1.25	1.85185185
-0.015	0.01433333	3.3333333	3.5416667	257.06667	0.003	-0.00566667	3.54166667	3.33333333
-0.0213333	0.008	8.3659228	5.625	245.73333	0.00766667	-0.009	5.625	8.51851852
-0.0276667	0.01266667	12.945212	6.4583333	254.4	0.01133333	-0.01033333	6.45833333	12.9452119
-0.029	0.007	36.560751	13.75	258.86667	0.018	-0.022	13.75	36.560751
-0.0376667	0.00666667	44.82619	16.458333	279.5	0.02033333	-0.02633333	16.4583333	44.8261896
-0.06	0.026	61.357067	19.375	307.16667	0.025	-0.031	19.375	61.357067
0.02666667	-0.053	74.345613	25	367.96667	0.02866667	-0.04	25	74.3456134
0.02233333	-0.0563333	83.791829	32.708333	339	0.03133333	-0.05233333	32.7083333	83.791829
0.02333333	-0.0696667	119.21514	41.041667	384.66667	0.04133333	-0.06566667	41.0416667	119.215138
-0.002	-0.0396667	139.28835	49.791667	378.3	0.047	-0.07966667	49.7916667	139.288346
-0.0156667	-0.044	48.36852	23.75	351.96667	0.02133333	-0.038	23.75	48.3685205
-0.017	-0.032	36.560751	17.5	355.06667	0.018	-0.028	17.5	36.560751
-0.0213333	-0.0286667	28.295312	15.416667	285.76667	0.01566667	-0.02466667	15.4166667	28.2953123
-0.0243333	-0.0253333	21.210651	13.333333	320.43333	0.01366667	-0.02133333	13.3333333	21.2106506
-0.024	-0.0256667	15.306766	14.791667	330.23333	0.012	-0.02366667	14.7916667	15.3067659
-0.024	-0.026	15.306766	15	236.23333	0.012	-0.024	15	15.3067659
-0.0256667	-0.022	12.945212	12.5	237.03333	0.01133333	-0.02	12.5	12.9452119
-0.026	-0.0236667	11.764435	12.916667	248.76667	0.011	-0.02066667	12.9166667	11.764435
-0.0266667	-0.018	9.4028811	9.375	233.4	0.01033333	-0.015	9.375	11.4814815
-0.028	-0.0186667	11.764435	9.7916667	267.13333	0.011	-0.01566667	9.79166667	12.2222222
-0.0256667	-0.0346667	27.114535	18.541667	254.06667	0.01533333	-0.02966667	18.5416667	27.1145354
-0.062	-0.0454667	25.933758	20.435417	309.4	0.015	-0.03269667	20.4354167	25.9337584
-0.0606667	-0.0460433	34.199197	21.570833	294.3	0.01733333	-0.03451333	21.5708333	34.1991971
-0.0606667	-0.0383533	41.283859	17.927083	302.7	0.01933333	-0.02868333	17.9270833	41.2838588
-0.061	-0.03836	29.476089	20.86875	248.9	0.016	-0.03339	20.86875	29.4760893
-0.0673333	0.00533333	4.0740741	5.4166667	305.23333	0.00366667	-0.00866667	5.41666667	4.07407407
-0.064	0.004	8.8888889	5.625	263.76667	0.008	-0.009	5.625	8.88888889
-0.0643333	0.00166667	9.6296296	7.0833333	278.13333	0.00866667	-0.01133333	7.08333333	9.62962963
-0.065	0.002	18.849097	7.5	315.5	0.013	-0.012	7.5	14.4444444
-0.0653333	-0.0103333	35.379974	15.208333	350.7	0.01766667	-0.02433333	15.2083333	19.6296296
-0.066	-0.018	40.103082	18.75	351.66667	0.019	-0.03	18.75	21.1111111
-0.0393333	-0.021	28.295312	20	372.03333	0.01566667	-0.032	20	28.2953123
-0.0343333	-0.0253333	60.17629	22.708333	317.23333	0.02466667	-0.03633333	22.7083333	60.17629
-0.0286667	-0.03	97.961152	26.875	362.46667	0.03533333	-0.043	26.875	97.9611524
-0.018	-0.039	142.83068 0	31.875	329.2	0.048	-0.051	31.875	142.830677
-0.0746667	0.01	5.9259259	3.125	301.33333	0.00533333	-0.005	3.125	5.92592593
-0.075	0.00933333	7.7777778	3.5416667	333.63333	0.007	-0.00566667	3.54166667	7.77777778
-0.0723333	0.001	14.125989	10	307.76667	0.01166667	-0.016	10	14.1259889
-0.079	0.00066667	18.849097	11.458333	298.56667	0.013	-0.01833333	11.4583333	18.8490967
-0.0816667	-0.0013333	23.572205	13.958333	305.83333	0.01433333	-0.02233333	13.9583333	23.5722045
-0.086	-0.0223333	29.476089	25.833333	373.93333	0.016	-0.04133333	25.8333333	29.4760893
-0.0903333	-0.017	35.379974	29.375	370.06667	0.01766667	-0.047	29.375	35.379974
-0.0913333	-0.0226667	60.17629	31.041667	379.23333	0.02466667	-0.04966667	31.0416667	60.17629
-0.097	-0.0286667	64.899398	37.291667	368.5	0.026	-0.05966667	37.2916667	64.8993978
-0.0983333	-0.032	81.430275	39.375	380.76667	0.03066667	-0.063	39.375	81.4302751
-0.103	-0.0333333	89.695714	42.708333	372	0.033	-0.06833333	42.7083333	89.6957138
0	0	0	0	0	0	0	0	

Table C.5: Calculated values for experimental variables

0.004	0.00866667	57.814736	18.333333	371.56667	0.024	-0.02933333	18.3333333	57.8147361
-0.009	0.017	47.187744	18.75	349.13333	0.021	-0.03	18.75	47.1877435
-0.017	0.01733333	47.187744	19.166667	389.2	0.021	-0.03066667	19.1666667	47.1877435
-0.0286667	0.02033333	48.36852	21.666667	345.73333	0.02133333	-0.03466667	21.6666667	48.3685205
-0.0353333	0.019	56.633959	23.75	434.76667	0.02366667	-0.038	23.75	56.6339591
-0.0426667	0.021	58.995513	23.125	327.43333	0.02433333	-0.037	23.125	58.995513
-0.0113333	-0.0413333	63.718621	19.583333	351.36667	0.02566667	-0.03133333	19.5833333	63.7186209
-0.024	-0.0423333	40.103082	22.708333	377.33333	0.019	-0.03633333	22.7083333	40.1030818
-0.0246667	-0.0466667	48.36852	24.166667	290.1	0.02133333	-0.03866667	24.1666667	48.3685205
-0.035	-0.034	36.560751	20	373.43333	0.018	-0.032	20	36.560751
-0.038	-0.0366667	40.103082	21.666667	356.16667	0.019	-0.03466667	21.6666667	40.1030818
-0.0423333	-0.0373333	49.549297	22.708333	308.8	0.02166667	-0.03633333	22.7083333	49.5492974
-0.0456667	-0.035049	51.910851	21.624375	329.9	0.02233333	-0.034599	21.624375	51.9108513
-0.049	-0.03474	61.357067	21.76875	296.3	0.025	-0.03483	21.76875	61.357067
-0.067	-0.01052	57.814736	6.5125	330.8	0.024	-0.01042	6.5125	26.6666667
-0.065	-0.01258	64.899398	7.51875	427.6	0.026	-0.01203	7.51875	28.8888889
-0.066	-0.01569	50.730074	9.53125	360.7	0.022	-0.01525	9.53125	24.4444444
-0.067	-0.0412	50.730074	25.6125	325.2	0.022	-0.04098	25.6125	24.4444444
-0.068	-0.01389	54.272405	8.54375	499.1	0.023	-0.01367	8.54375	25.5555556

Table C.6: Calculated values for experimental variables

C2 TEST SAMPLES

The data shown above is an excel spreadsheet representation of the actual experimental results. The values displayed in the frequency and force columns are all averages over 10000 sample points. Given the large number of sample points, this information will be displayed on the attached Compact Disk (CD).

APPENDIX D

QUALITATIVE AND QUANTITATIVE ANALYSIS

- Qualitative data for comparison
- Additional qualitative charts
- Quantitative data for comparison
- Summary of quantitative trends

D1 QUALITATIVE DATA

FEED

PF tool

feed	Actual Feed	Actual Tang	feed	Frequency	feed	Ra	Ry
0.03	3.125	4.07407407	0.03	293.9333	0.03	2.54	17.8
0.04	6.25	5.18518519	0.04	304.9333	0.04	2.7	19.8
0.05	6.666666667	8.22210414	0.05	302.1333	0.05	2.56	17.3
0.06	8.541666667	17.6683198	0.06	305.8333	0.06	2.72	17
0.08	8.958333333	28.2953123	0.08	308.2	0.08	3.13	20.6
0.1	11.45833333	29.4760893	0.1	327.1667	0.1	2.99	18.9
0.12	13.54166667	42.4646357	0.12	317.1	0.12	3.32	23.2
0.16	16.04166667	64.8993978	0.16	323.3	0.16	3.47	19.4
0.2	21.66666667	133.384461	0.2	359.8333	0.2	3.67	22.6
0.25	27	226.66584	0.25	365.2667	0.25	3.88	21.9

PM tool

feed	Actual Feed	Actual Tang	feed	Frequency	feed	Ra	Ry
0.03	5.416666667	4.07407407	0.03	305.2333	0.03	2.66	16.3
0.04	5.625	8.88888889	0.04	263.7667	0.04	2.76	17.4
0.05	7.083333333	9.62962963	0.05	278.1333	0.05	2.85	18.3
0.06	7.5	14.4444444	0.06	315.5	0.06	3.07	20.2
0.08	15.20833333	19.6296296	0.08	350.7	0.08	3.01	19.9
0.1	18.75	21.1111111	0.1	351.6667	0.1	3.33	24.8
0.12	20	28.2953123	0.12	372.0333	0.12	3.42	27
0.16	22.70833333	60.17629	0.16	317.2333	0.16	3.23	23.5
0.2	26.875	97.9611524	0.2	362.4667	0.2	3.44	28.4
0.25	31.875	142.830677	0.25	329.2	0.25	3.644	27.7

Table D.1: Qualitative data for varying feed

SPEED

PF Tool

speed	Ffeed	Ftang	speed	freq	speed	Ra	Ry
70	23.75	48.3685205	70	351.9667	70	4.2	38.7
160	17.5	36.560751	160	355.0667	160	3.41	28.8
250	15.41666667	28.2953123	250	285.7667	250	3.7	22.6
340	13.33333333	21.2106506	340	320.4333	340	3.76	33.1
430	14.79166667	15.3067659	430	330.2333	430	2.58	15
520	15	15.3067659	520	236.2333	520	3.09	20.8
610	12.5	12.9452119	610	237.0333	610	2.43	15.6
700	12.91666667	11.764435	700	248.7667	700	2.71	17.3
790	9.375	11.4814815	790	233.4	790	2.52	15.8
880	9.791666667	12.2222222	880	267.1333	880	2.78	16.8
970	18.54166667	27.1145354	970	254.0667	970	2.23	14
1000	20.43541667	25.9337584	1000	309.4	1000	2.46	13.8
1100	21.57083333	34.1991971	1100	294.3	1100	2.64	16.5
1200	17.92708333	41.2838588	1200	302.7	1200	1.57	11.2
1300	20.86875	29.4760893	1300	248.9	1300	1.82	9.9

PM Tool

Speed	Ffeed	Ftang	Speed	Frequency	Speed	Ra	Ry
160	18.33333333	57.8147361	160	371.5667	160	4.67	32.8
250	18.75	47.1877435	250	349.1333	250	3.85	27.6
340	19.16666667	47.1877435	340	389.2	340	3.73	25.9
430	21.66666667	48.3685205	430	345.7333	430	4.38	25.8
520	23.75	56.6339591	520	434.7667	520	3.54	22.4
610	23.125	58.995513	610	327.4333	610	3.28	25.2
700	19.58333333	57.6	700	351.3667	700	3.33	21.3
790	22.70833333	40.1030818	790	377.3333	790	3.13	21.1
880	24.16666667	48.3685205	880	290.1	880	2.2	19.5
970	20	45.2	970	373.4333	970	1.94	13.6
1000	21.66666667	40.1030818	1000	356.1667	1000	1.54	14.9
1100	22.70833333	49.5492974	1100	308.8	1100	1.47	9.7
1200	21.624375	51.9108513	1200	329.9	1200	1.32	10.3
1300	21.76875	54.5	1300	296.3	1300	1.22	11.7

Table D.2: Qualitative data for varying speed

DEPTH

PF Tool

depth	Ffeed	Ftang	depth	Freq	depth	Ra	Ry
0.1	1.25	1.85185185	0.1	168.1	0.1	3.19	20.4
0.2	3.541666667	3.33333333	0.2	257.0667	0.2	3.33	21.2
0.3	5.625	8.51851852	0.3	245.7333	0.3	3.03	19.4
0.4	6.458333333	12.9452119	0.4	254.4	0.4	3.06	19.6
0.5	13.75	36.560751	0.5	258.8667	0.5	3.16	22.2
0.6	16.45833333	44.8261896	0.6	279.5	0.6	3.38	25.6
0.7	19.375	61.357067	0.7	307.1667	0.7	3.45	26.3
0.8	25	74.3456134	0.8	367.9667	0.8	3.75	28.6
0.9	32.70833333	83.791829	0.9	339	0.9	3.88	25
1	41.04166667	119.215138	1	384.6667	1	3.78	26.1
1.1	49.79166667	139.288346	1.1	378.3	1.1	3.84	27

PM Tool

depth	Ffeed	Ftang	depth	Freq	depth	Ra	Ry
0.1	3.125	5.92592593	0.1	301.3333	0.1	3.06	19.6
0.2	3.541666667	7.77777778	0.2	333.6333	0.2	3.39	24.13
0.3	10	14.1259889	0.3	307.7667	0.3	3.34	25.65
0.4	11.45833333	18.8490967	0.4	298.5667	0.4	3.35	24.2
0.5	13.95833333	23.5722045	0.5	305.8333	0.5	3.42	23
0.6	25.83333333	29.4760893	0.6	373.9333	0.6	3.46	20.9
0.7	29.375	35.379974	0.7	370.0667	0.7	3.54	20
0.8	31.04166667	60.17629	0.8	379.2333	0.8	3.59	22.9
0.9	37.29166667	64.8993978	0.9	368.5	0.9	3.67	21.9
1	39.375	81.4302751	1	380.7667	1	3.7	29.7
1.1	42.70833333	89.6957138	1.1	372	1.1	3.85	31.12

Table D.3: Qualitative data for varying depth of cut

D2 QUALITATIVE ANALYSIS

These figures show the continued qualitative trends for a PM type tool. The PR diagrams are displayed within the thesis (chapter 6).

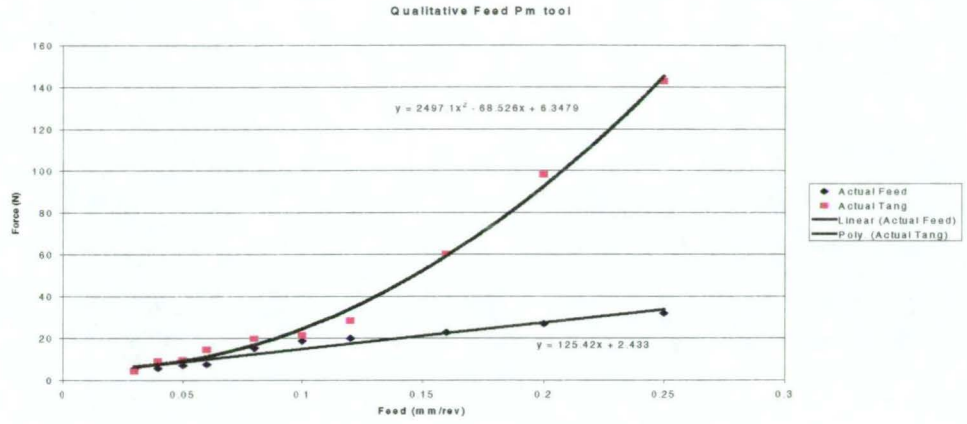

Figure D.1: Effect of feed rate on forces

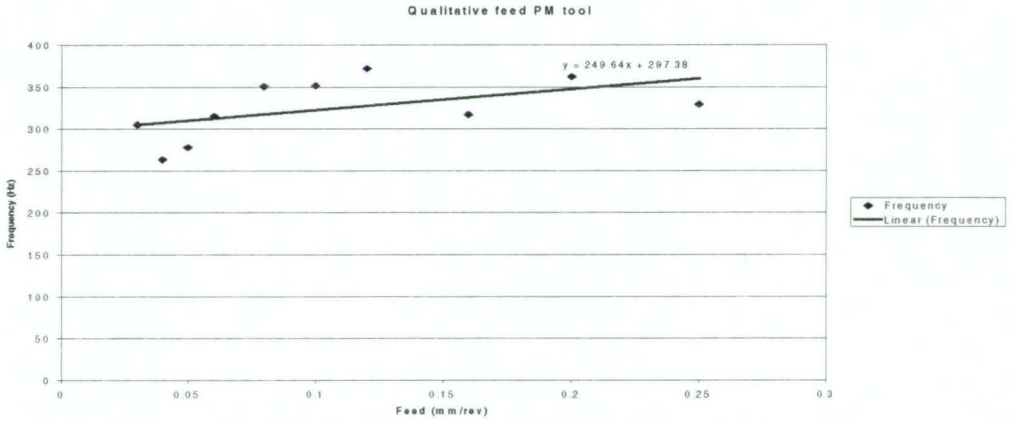

Figure D.2: Effect of feed rate on frequency

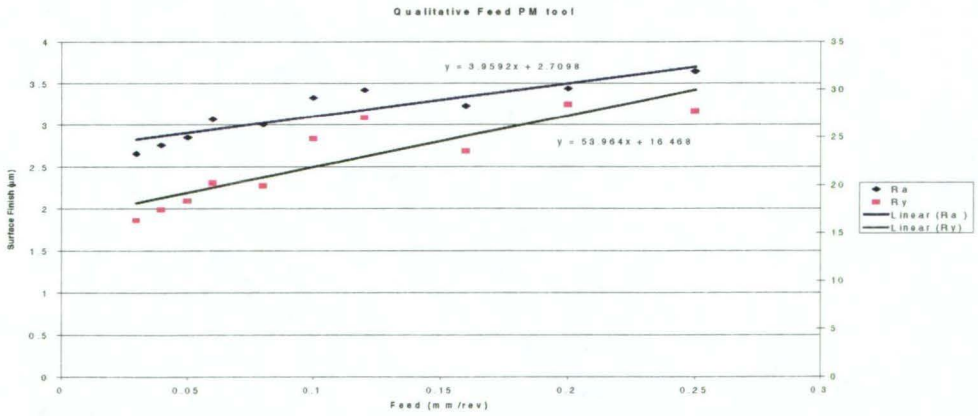

Figure D.3: Effect of feed rate on surface finish

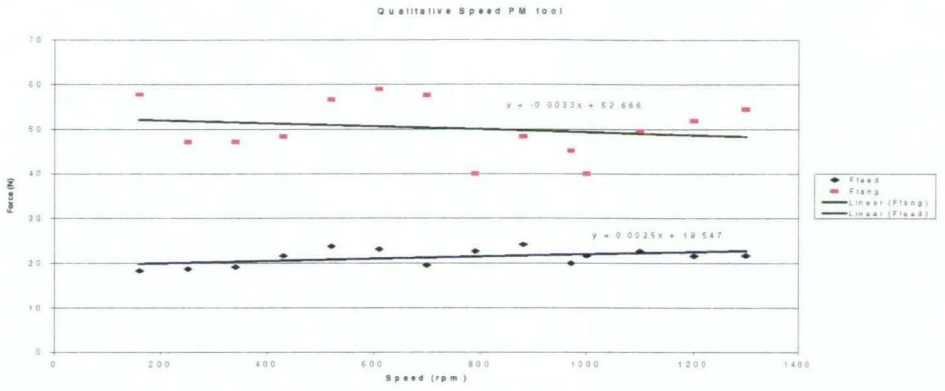

Figure D.4: Effect of speed rate on forces

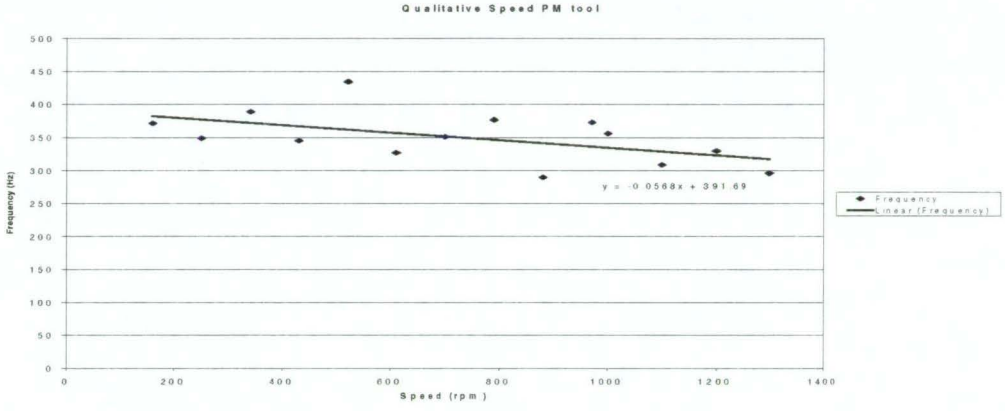

Figure D.5: Effect of speed rate on frequency

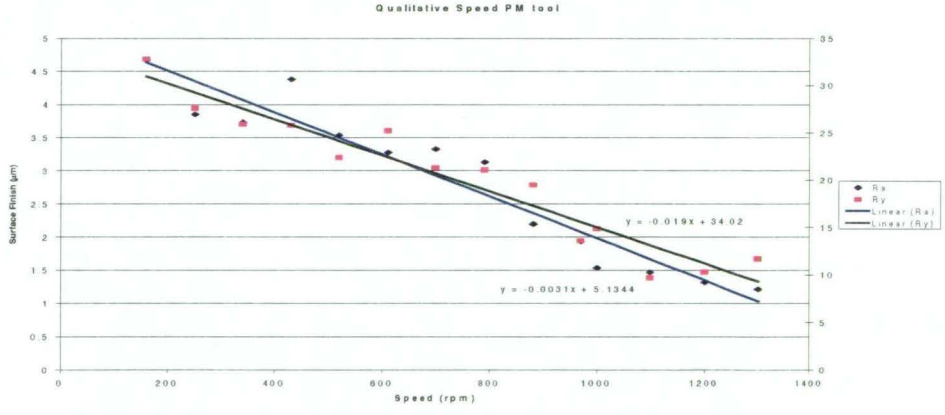

Figure D.6: Effect of speed rate on surface finish

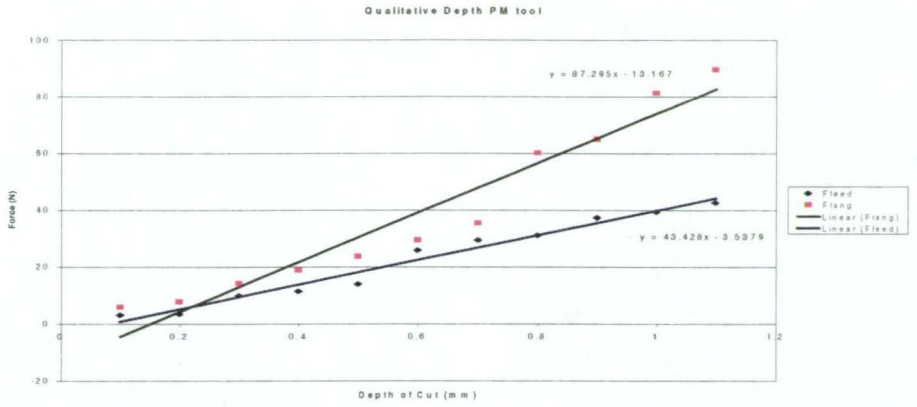

Figure D.7: Effect of depth of cut on forces

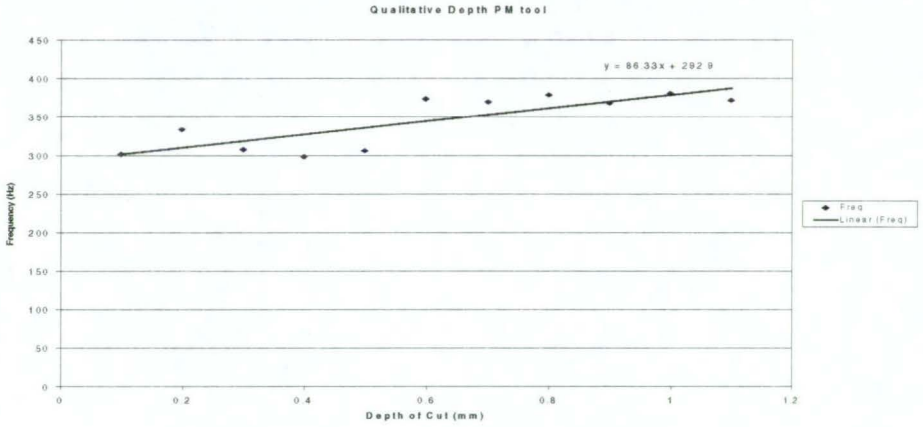

Figure D8: Effect of depth of cut on frequency

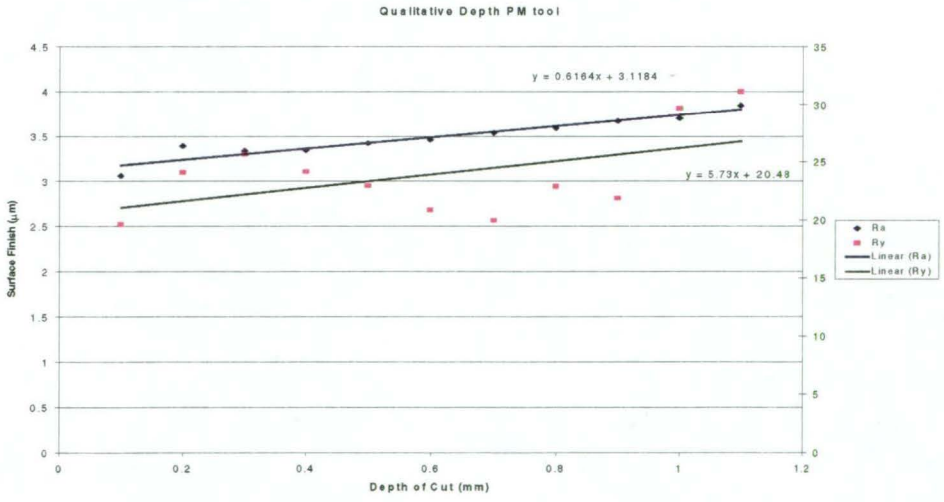

Figure D9: Effect of depth of cut on surface finish

D3 QUANTITATIVE DATA

FEED

feed	PF tool Feed	PF tool Tang	PM tool Feed	PM tool Tang	feed	PF tool Freq	PM tool Freq
0.03	3.125	4.074074074	5.416666667	4.074074074	0.03	293.93333	305.233333
0.04	6.25	5.185185185	5.625	8.888888889	0.04	304.93333	263.766667
0.05	6.666666667	8.222104145	7.083333333	9.62962963	0.05	302.13333	278.133333
0.06	8.541666667	17.66831975	7.5	14.44444444	0.06	305.83333	315.5
0.08	8.958333333	28.29531232	15.20833333	19.62962963	0.08	308.2	350.7
0.1	11.45833333	29.47608927	18.75	21.11111111	0.1	327.16667	351.666667
0.12	13.54166667	42.46463573	20	28.29531232	0.12	317.1	372.033333
0.16	16.04166667	64.8993978	22.70833333	60.17629	0.16	323.3	317.233333
0.2	21.66666667	133.384461	26.875	97.96115244	0.2	359.83333	362.466667
0.25	27	226.6658401	31.875	142.8306766	0.25	365.26667	329.2

feed	PF tool Ra	PM tool Ra	feed	PF tool Ry	PM tool Ry
0.03	2.54	2.66	0.03	17.8	16.3
0.04	2.7	2.76	0.04	19.8	17.4
0.05	2.56	2.85	0.05	17.3	18.3
0.06	2.72	3.07	0.06	17	20.2
0.08	3.13	3.01	0.08	20.6	19.9
0.1	2.99	3.33	0.1	18.9	24.8
0.12	3.32	3.42	0.12	23.2	27
0.16	3.47	3.23	0.16	19.4	23.5
0.2	3.67	3.44	0.2	22.6	28.4
0.25	3.88	3.644	0.25	21.9	27.7

Table D.4: Quantitative data for varying feed

SPEED

Speed	PF tool Feed	PM tool Feed	Speed	PF tool Tang	PM tool Tang	Speed	PF tool Freq	PM tool Freq
160	17.5	18.33333333	160	36.56075097	57.8147361	160	355.066667	371.566667
250	15.41666667	18.75	250	28.29531232	47.18774354	250	285.766667	349.133333
340	13.33333333	19.16666667	340	21.21065061	47.18774354	340	320.433333	389.2
430	14.79166667	21.66666667	430	15.30676585	48.36852049	430	330.233333	345.733333
520	15	23.75	520	15.30676585	56.63395915	520	236.233333	434.766667
610	12.5	23.125	610	12.94521195	58.99551305	610	237.033333	327.433333
700	12.91666667	19.58333333	700	11.764435	57.6	700	248.766667	351.366667
790	9.375	22.70833333	790	11.48148148	40.10308183	790	233.4	377.333333
880	9.791666667	24.16666667	880	12.22222222	48.36852049	880	267.133333	290.1
970	18.54166667	20	970	27.11453536	45.2	970	254.066667	373.433333
1000	20.43541667	21.66666667	1000	25.93375841	40.10308183	1000	309.4	356.166667
1100	21.57083333	22.70833333	1100	34.19919707	49.54929744	1100	294.3	308.8
1200	17.92708333	21.624375	1200	41.28385878	51.91085134	1200	302.7	329.9
1300	20.86875	21.76875	1300	29.47608927	54.5	1300	248.9	296.3

Speed	PF tool Ra	PM tool Ra	Speed	PF tool Ry	PM tool Ry
160	3.41	4.67	160	28.8	32.8
250	3.7	3.85	250	22.6	27.6
340	3.76	3.73	340	33.1	25.9
430	2.58	4.38	430	15	25.8
520	3.09	3.54	520	20.8	22.4
610	2.43	3.28	610	15.6	25.2
700	2.71	3.33	700	17.3	21.3
790	2.52	3.13	790	15.8	21.1
880	2.78	2.2	880	16.8	19.5
970	2.23	1.94	970	14	13.6
1000	2.46	1.54	1000	13.8	14.9
1100	2.64	1.47	1100	16.5	9.7
1200	1.57	1.32	1200	11.2	10.3
1300	1.82	1.22	1300	9.9	11.7

Table D.5: Quantitative data for varying speed

DEPTH

depth	PF tool Feed	PM tool Feed	depth	PF tool Tang	PM tool Tang	depth	PF tool Freq	PM tool Freq
0.1	1.25	3.125	0.1	1.851851852	5.925925926	0.1	168.1	301.333333
0.2	3.541666667	3.541666667	0.2	3.333333333	7.777777778	0.2	257.066667	333.633333
0.3	5.625	10	0.3	8.518518519	14.1259889	0.3	245.733333	307.766667
0.4	6.458333333	11.45833333	0.4	12.94521195	18.84909671	0.4	254.4	298.566667
0.5	13.75	13.95833333	0.5	36.56075097	23.57220451	0.5	258.866667	305.833333
0.6	16.45833333	25.83333333	0.6	44.82618963	29.47608927	0.6	279.5	373.933333
0.7	19.375	29.375	0.7	61.35706695	35.37997402	0.7	307.166667	370.066667
0.8	25	31.04166667	0.8	74.34561341	60.17629	0.8	367.966667	379.233333
0.9	32.70833333	37.29166667	0.9	83.79182902	64.8993978	0.9	339	368.5
1	41.04166667	39.375	1	119.2151376	81.43027512	1	384.666667	380.766667
1.1	49.79166667	42.70833333	1.1	139.2883457	89.69571378	1.1	378.3	372

Speed	PF tool Ra	PM tool Ra	Speed	PF tool Ry	PM tool Ry
0.1	3.19	3.06	0.1	20.4	19.6
0.2	3.33	3.39	0.2	21.2	24.13
0.3	3.03	3.34	0.3	19.4	25.65
0.4	3.06	3.35	0.4	19.6	24.2
0.5	3.16	3.42	0.5	22.2	23
0.6	3.38	3.46	0.6	25.6	20.9
0.7	3.45	3.54	0.7	26.3	20
0.8	3.75	3.59	0.8	28.6	22.9
0.9	3.88	3.67	0.9	25	21.9
1	3.78	3.7	1	26.1	29.7
1.1	3.84	3.85	1.1	27	31.12

Table D.6: Quantitative data for varying depth of cut

D4 SUMMARY OF QUANTITATIVE TRENDS

The quantitative trend charts are shown in chapter 6.2. The following tables are a summary of the regression, or best fit lines.

VARIABLE	TOOL	OUTPUT	REGRESSION LINE
Feed	PF	$F_{\tan g}$	$y = 19696x^3 - 3307.6x^2 + 553.27x - 10.701$
Feed	PM	$F_{\tan g}$	$y = -3951.2x^3 + 4.131.2x^2 - 256.94 + 11.918$
Feed	PF	F_{feed}	y = 100.49x + 1.3716
Feed	PM	F_{feed}	y = 125.42x + 2.433
Feed	PF	f	y = 313.59x + 286.59
Feed	PM	f	y = 249.64x + 297.38
Feed	PF	R_a	y = 6.276x + 2.4139
Feed	PM	R_a	y = 3.9592x + 2.7098
Feed	PF	R_y	y = 20.446x + 17.621
Feed	PM	R_y	y = 53.964x + 16.468

Table D.7: Summary of Quantitative feed regression lines

VARIABLE	TOOL	OUTPUT	REGRESSION LINE
Speed	PF	$F_{\tan g}$	y = 0.007x + 17.988
Speed	PM	$F_{\tan g}$	y = -0.0033x + 52.666
Speed	PF	F_{feed}	y = 0.0046x + 12.354
Speed	PM	F_{feed}	y = 0.0025x + 19.547
Speed	PF	f	y = -0.0389x + 308.69
Speed	PM	f	y = -0.0568x + 391.69
Speed	PF	R_a	y = -0.0015x + 3.7908
Speed	PM	R_a	y = -0.0031x + 5.1344

Speed	PF	R_y	y = -0.0144x + 28.478
Speed	PM	R_y	y = -0.019x + 34.02

Table D.8: Summary of Quantitative speed regression lines

VARIABLE	TOOL	OUTPUT	REGRESSION LINE
Depth	PF	F_{tang}	Y = 138.58x – 29.858
Depth	PM	F_{tang}	Y = 87.295x – 13.167
Depth	PF	F_{feed}	Y = 46.97x – 8.634
Depth	PM	F_{feed}	Y = 43.428x – 3.5379
Depth	PF	f	Y = 192.42x + 179.16
Depth	PM	f	Y = 86.33x + 292.9
Depth	PF	R_a	Y = 0.8427x + 2.9353
Depth	PM	R_a	Y = 0.6164x + 3.1184
Depth	PF	R_y	Y = 8.3182x + 18.773
Depth	PM	R_y	Y = 5.73x + 20.48

Table D.9: Summary of Quantitative depth of cut regression lines

APPENDIX E

BACK PROPAGATION NEURAL NETWORK DATA

- Raw neural network data
- BPNN data for surface finish prediction
- BPNN normalised data for surface finish prediction

E1 COMPLETE NEURAL NETWORK DATA

FEED	SPEED	DEPTH	Tool Type	Tangential Force	Feed Force	Freq	Ra	Ry
0.03	300.00	0.50	0.00	5.56	4.38	270.00	2.54	17.80
0.04	300.00	0.50	0.00	5.56	7.50	310.60	2.70	19.80
0.05	300.00	0.50	0.00	8.22	6.88	310.60	2.56	17.30
0.06	300.00	0.50	0.00	22.39	8.75	331.40	2.72	17.00
0.08	300.00	0.50	0.00	36.56	8.75	288.60	3.13	20.60
0.10	300.00	0.50	0.00	33.02	12.50	292.00	2.99	18.90
0.12	300.00	0.50	0.00	43.65	13.75	335.60	3.32	23.20
0.16	300.00	0.50	0.00	64.90	16.25	328.30	3.47	19.40
0.20	300.00	0.50	0.00	135.75	21.25	361.70	3.67	22.60
0.25	300.00	0.50	0.00	231.39	26.38	357.20	3.88	21.90
0.03	300.00	0.50	0.00	3.33	2.50	309.40	2.54	17.80
0.04	300.00	0.50	0.00	3.33	7.50	350.10	2.70	19.80
0.05	300.00	0.50	0.00	8.22	6.25	275.80	2.56	17.30
0.06	300.00	0.50	0.00	18.85	9.38	250.90	2.72	17.00
0.08	300.00	0.50	0.00	29.48	8.75	315.20	3.13	20.60
0.10	300.00	0.50	0.00	29.48	11.25	369.40	2.99	18.90
0.12	300.00	0.50	0.00	43.65	13.13	335.90	3.32	23.20
0.16	300.00	0.50	0.00	64.90	15.63	321.10	3.47	19.40
0.20	300.00	0.50	0.00	132.20	21.88	361.60	3.67	22.60
0.25	300.00	0.50	0.00	224.30	27.63	377.90	3.88	21.90
0.03	300.00	0.50	0.00	3.33	2.50	302.40	2.54	17.80
0.04	300.00	0.50	0.00	6.67	3.75	254.10	2.70	19.80
0.05	300.00	0.50	0.00	8.22	6.88	320.00	2.56	17.30
0.06	300.00	0.50	0.00	11.76	7.50	335.20	2.72	17.00
0.08	300.00	0.50	0.00	18.85	9.38	320.80	3.13	20.60
0.10	300.00	0.50	0.00	25.93	10.63	320.10	2.99	18.90
0.12	300.00	0.50	0.00	40.10	13.75	279.80	3.32	23.20
0.16	300.00	0.50	0.00	64.90	16.25	320.50	3.47	19.40
0.20	300.00	0.50	0.00	132.20	21.88	356.20	3.67	22.60
0.25	300.00	0.50	0.00	224.30	27.00	360.70	3.88	21.90
0.08	300.00	0.10	0.00	2.22	2.50	166.30	3.19	20.40
0.08	300.00	0.20	0.00	5.56	3.75	275.20	3.33	21.20
0.08	300.00	0.30	0.00	11.76	6.25	244.00	3.03	19.40
0.08	300.00	0.40	0.00	15.31	6.88	248.20	3.06	19.60
0.08	300.00	0.50	0.00	43.65	13.75	270.90	3.16	22.20
0.08	300.00	0.60	0.00	50.73	16.88	272.50	3.38	25.60
0.08	300.00	0.70	0.00	61.36	19.38	348.20	3.45	26.30
0.08	300.00	0.80	0.00	75.53	26.25	379.00	3.75	28.60
0.08	300.00	0.90	0.00	82.61	31.88	365.00	3.88	25.00
0.08	300.00	1.00	0.00	121.58	39.38	375.00	3.78	26.10
0.08	300.00	1.10	0.00	139.29	47.50	409.80	3.84	27.00
0.08	300.00	0.10	0.00	3.33	1.25	164.80	3.19	20.40
0.08	300.00	0.20	0.00	2.22	3.13	250.60	3.33	21.20
0.08	300.00	0.30	0.00	7.78	4.38	268.80	3.03	19.40
0.08	300.00	0.40	0.00	15.31	6.25	244.80	3.06	19.60
0.08	300.00	0.50	0.00	36.56	13.13	252.10	3.16	22.20
0.08	300.00	0.60	0.00	43.65	16.25	214.00	3.38	25.60
0.08	300.00	0.70	0.00	64.90	19.38	241.20	3.45	26.30
0.08	300.00	0.80	0.00	75.53	23.75	354.20	3.75	28.60
0.08	300.00	0.90	0.00	86.15	33.13	336.70	3.88	25.00
0.08	300.00	1.00	0.00	118.03	41.25	404.50	3.78	26.10
0.08	300.00	1.10	0.00	142.83	50.63	365.00	3.84	27.00
0.08	300.00	0.10	0.00	0.00	0.00	173.20	3.19	20.40
0.08	300.00	0.20	0.00	2.22	3.75	245.40	3.33	21.20
0.08	300.00	0.30	0.00	5.56	6.25	224.40	3.03	19.40
0.08	300.00	0.40	0.00	8.22	6.25	270.20	3.06	19.60
0.08	300.00	0.50	0.00	29.48	14.38	253.60	3.16	22.20
0.08	300.00	0.60	0.00	40.10	16.25	352.00	3.38	25.60

Table E.1: Complete BPNN training data

0.08	300.00	0.70	0.00	57.81	19.38	332.10	3.45	26.30
0.08	300.00	0.80	0.00	71.98	25.00	370.70	3.75	28.60
0.08	300.00	0.90	0.00	82.61	33.13	315.30	3.88	25.00
0.08	300.00	1.00	0.00	118.03	42.50	374.50	3.78	26.10
0.08	300.00	1.10	0.00	135.75	51.25	360.10	3.84	27.00
0.08	70.00	0.50	0.00	47.19	23.75	350.40	4.20	38.70
0.08	160.00	0.50	0.00	36.56	17.50	342.00	3.41	28.80
0.08	250.00	0.50	0.00	25.93	15.00	344.20	3.70	22.60
0.08	340.00	0.50	0.00	18.85	13.13	356.90	3.76	33.10
0.08	430.00	0.50	0.00	15.31	14.38	352.70	2.58	15.00
0.08	520.00	0.50	0.00	15.31	14.38	204.10	3.09	20.80
0.08	610.00	0.50	0.00	15.31	12.50	231.40	2.43	15.60
0.08	700.00	0.50	0.00	11.76	13.13	238.20	2.71	17.30
0.08	790.00	0.50	0.00	8.22	9.38	299.90	2.52	15.80
0.08	880.00	0.50	0.00	11.76	9.38	256.40	2.78	16.80
0.08	970.00	0.50	0.00	22.39	18.75	265.50	2.23	14.00
0.08	1000.00	0.50	0.00	29.48	20.28	302.90	2.46	13.80
0.08	1100.00	0.50	0.00	36.56	21.39	272.30	2.64	16.50
0.08	1200.00	0.50	0.00	40.10	17.88	281.70	1.57	11.20
0.08	1300.00	0.50	0.00	29.48	20.86	245.50	1.82	9.90
0.08	70.00	0.50	0.00	47.19	23.75	374.90	4.20	38.70
0.08	160.00	0.50	0.00	36.56	16.88	360.00	3.41	28.80
0.08	250.00	0.50	0.00	25.93	15.63	263.30	3.70	22.60
0.08	340.00	0.50	0.00	22.39	13.75	364.20	3.76	33.10
0.08	430.00	0.50	0.00	15.31	15.00	317.20	2.58	15.00
0.08	520.00	0.50	0.00	15.31	15.63	243.90	3.09	20.80
0.08	610.00	0.50	0.00	11.76	12.50	226.10	2.43	15.60
0.08	700.00	0.50	0.00	11.76	11.88	279.70	2.71	17.30
0.08	790.00	0.50	0.00	8.22	8.75	200.20	2.52	15.80
0.08	880.00	0.50	0.00	11.76	9.38	305.00	2.78	16.80
0.08	970.00	0.50	0.00	29.48	18.75	266.90	2.23	14.00
0.08	1000.00	0.50	0.00	25.93	20.74	261.90	2.46	13.80
0.08	1100.00	0.50	0.00	33.02	21.94	300.30	2.64	16.50
0.08	1200.00	0.50	0.00	43.65	18.02	281.40	1.57	11.20
0.08	1300.00	0.50	0.00	29.48	20.89	210.80	1.82	9.90
0.08	70.00	0.50	0.00	50.73	23.75	330.60	4.20	38.70
0.08	160.00	0.50	0.00	36.56	18.13	363.20	3.41	28.80
0.08	250.00	0.50	0.00	33.02	15.63	249.80	3.70	22.60
0.08	340.00	0.50	0.00	22.39	13.13	240.20	3.76	33.10
0.08	430.00	0.50	0.00	15.31	15.00	320.80	2.58	15.00
0.08	520.00	0.50	0.00	15.31	15.00	260.70	3.09	20.80
0.08	610.00	0.50	0.00	11.76	12.50	253.60	2.43	15.60
0.08	700.00	0.50	0.00	11.76	13.75	228.40	2.71	17.30
0.08	790.00	0.50	0.00	11.76	10.00	200.10	2.52	15.80
0.08	880.00	0.50	0.00	11.76	10.63	240.00	2.78	16.80
0.08	970.00	0.50	0.00	29.48	18.13	229.80	2.23	14.00
0.08	1000.00	0.50	0.00	22.39	20.28	363.40	2.46	13.80
0.08	1100.00	0.50	0.00	33.02	21.39	310.30	2.64	16.50
0.08	1200.00	0.50	0.00	40.10	17.88	345.00	1.57	11.20
0.08	1300.00	0.50	0.00	29.48	20.86	290.40	1.82	9.90
0.03	300.00	0.50	1.00	4.44	5.63	274.10	2.66	16.30
0.04	300.00	0.50	1.00	8.89	5.63	220.50	2.76	17.40
0.05	300.00	0.50	1.00	8.89	6.25	304.00	2.85	18.30
0.06	300.00	0.50	1.00	18.85	7.50	295.40	3.07	20.20
0.08	300.00	0.50	1.00	36.56	15.63	357.30	3.01	19.90
0.10	300.00	0.50	1.00	40.10	18.75	342.00	3.33	24.80
0.12	300.00	0.50	1.00	33.02	20.63	360.60	3.42	27.00
0.16	300.00	0.50	1.00	61.36	21.88	267.80	3.23	23.50
0.20	300.00	0.50	1.00	96.78	26.88	378.00	3.44	28.40
0.25	300.00	0.50	1.00	142.83	31.25	336.30	3.64	27.70

Table E.2: Complete BPNN training data

0.03	300.00	0.50	1.00	3.33	5.00	316.80	2.66	16.30
0.04	300.00	0.50	1.00	8.89	5.63	310.70	2.76	17.40
0.05	300.00	0.50	1.00	10.00	7.50	249.90	2.85	18.30
0.06	300.00	0.50	1.00	18.85	7.50	330.60	3.07	20.20
0.08	300.00	0.50	1.00	40.10	15.00	384.60	3.01	19.90
0.10	300.00	0.50	1.00	40.10	18.75	367.10	3.33	24.80
0.12	300.00	0.50	1.00	29.48	20.00	369.80	3.42	27.00
0.16	300.00	0.50	1.00	61.36	22.50	326.30	3.23	23.50
0.20	300.00	0.50	1.00	100.32	26.88	328.40	3.44	28.40
0.25	300.00	0.50	1.00	142.83	32.50	339.30	3.64	27.70
0.03	300.00	0.50	1.00	4.44	5.63	324.80	2.66	16.30
0.04	300.00	0.50	1.00	8.89	5.63	260.10	2.76	17.40
0.05	300.00	0.50	1.00	10.00	7.50	280.50	2.85	18.30
0.06	300.00	0.50	1.00	18.85	7.50	320.50	3.07	20.20
0.08	300.00	0.50	1.00	29.48	15.00	310.20	3.01	19.90
0.10	300.00	0.50	1.00	40.10	18.75	345.90	3.33	24.80
0.12	300.00	0.50	1.00	22.39	19.38	385.70	3.42	27.00
0.16	300.00	0.50	1.00	57.81	23.75	357.60	3.23	23.50
0.20	300.00	0.50	1.00	96.78	26.88	381.00	3.44	28.40
0.25	300.00	0.50	1.00	142.83	31.88	312.00	3.64	27.70
0.08	300.00	0.10	1.00	5.56	3.13	359.70	3.06	19.60
0.08	300.00	0.20	1.00	7.78	3.75	345.60	3.39	27.30
0.08	300.00	0.30	1.00	15.31	10.00	259.90	3.34	29.00
0.08	300.00	0.40	1.00	22.39	13.13	304.00	3.35	24.20
0.08	300.00	0.50	1.00	22.39	15.00	368.10	3.42	23.00
0.08	300.00	0.60	1.00	36.56	25.63	368.00	3.46	20.90
0.08	300.00	0.70	1.00	40.10	28.75	374.70	3.54	20.00
0.08	300.00	0.80	1.00	57.81	30.63	378.60	3.59	22.90
0.08	300.00	0.90	1.00	68.44	36.25	380.60	3.67	21.90
0.08	300.00	1.00	1.00	82.61	39.38	381.40	3.70	29.70
0.08	300.00	1.10	1.00	89.70	41.88	376.80	3.85	34.40
0.08	300.00	0.10	1.00	6.67	3.13	262.00	3.06	19.60
0.08	300.00	0.20	1.00	7.78	3.75	366.50	3.39	27.30
0.08	300.00	0.30	1.00	15.31	9.38	346.90	3.34	29.00
0.08	300.00	0.40	1.00	18.85	11.25	297.80	3.35	24.20
0.08	300.00	0.50	1.00	22.39	13.13	297.20	3.42	23.00
0.08	300.00	0.60	1.00	29.48	25.63	376.10	3.46	20.90
0.08	300.00	0.70	1.00	29.48	30.00	377.60	3.54	20.00
0.08	300.00	0.80	1.00	64.90	31.25	379.40	3.59	22.90
0.08	300.00	0.90	1.00	64.90	37.50	374.60	3.67	21.90
0.08	300.00	1.00	1.00	82.61	39.38	381.50	3.70	29.70
0.08	300.00	1.10	1.00	89.70	43.13	370.60	3.85	34.40
0.08	300.00	0.10	1.00	5.56	3.13	282.30	3.06	19.60
0.08	300.00	0.20	1.00	7.78	3.13	288.80	3.39	27.30
0.08	300.00	0.30	1.00	11.76	10.63	316.50	3.34	29.00
0.08	300.00	0.40	1.00	15.31	10.00	293.90	3.35	24.20
0.08	300.00	0.50	1.00	25.93	13.75	252.20	3.42	23.00
0.08	300.00	0.60	1.00	22.39	26.25	377.70	3.46	20.90
0.08	300.00	0.70	1.00	36.56	29.38	357.90	3.54	20.00
0.08	300.00	0.80	1.00	57.81	31.25	379.70	3.59	22.90
0.08	300.00	0.90	1.00	61.36	38.13	350.30	3.67	21.90
0.08	300.00	1.00	1.00	79.07	39.38	379.40	3.70	29.70
0.08	300.00	1.10	1.00	89.70	43.13	368.60	3.85	34.40
0.08	160.00	0.50	1.00	64.90	19.38	370.60	4.67	32.80
0.08	250.00	0.50	1.00	54.27	18.75	359.50	3.85	27.60
0.08	340.00	0.50	1.00	54.27	19.38	398.20	3.73	25.90
0.08	430.00	0.50	1.00	54.27	21.25	366.20	4.38	25.80
0.08	520.00	0.50	1.00	61.36	23.75	452.90	3.54	22.40
0.08	610.00	0.50	1.00	61.36	23.13	485.70	3.28	25.20
0.08	700.00	0.50	1.00	64.90	19.38	154.40	3.33	21.30

Table E.3: Complete BPNN training data

0.08	790.00	0.50	1.00	43.65	22.50	390.10	3.13	21.10
0.08	880.00	0.50	1.00	50.73	24.38	499.00	2.20	19.50
0.08	970.00	0.50	1.00	40.10	20.00	376.20	1.94	13.60
0.08	1000.00	0.50	1.00	43.65	21.88	362.30	1.54	14.90
0.08	1100.00	0.50	1.00	54.27	22.50	373.30	1.47	9.70
0.08	1200.00	0.50	1.00	54.27	21.69	317.50	1.32	10.30
0.08	1300.00	0.50	1.00	61.36	21.77	296.30	1.22	11.70
0.08	520.00	0.50	1.00	57.81	6.51	330.80	3.54	22.40
0.08	700.00	0.50	1.00	64.90	7.52	427.60	3.93	28.30
0.08	880.00	0.50	1.00	50.73	9.53	360.70	2.20	19.50
0.08	1060.00	0.50	1.00	50.73	25.61	325.20	1.14	7.90
0.08	1200.00	0.50	1.00	54.27	8.54	499.10	1.32	10.30
0.08	160.00	0.50	1.00	57.81	18.13	365.70	4.67	32.80
0.08	250.00	0.50	1.00	50.73	18.75	349.60	3.85	27.60
0.08	340.00	0.50	1.00	47.19	19.38	408.70	3.73	25.90
0.08	430.00	0.50	1.00	47.19	21.88	265.50	4.38	25.80
0.08	520.00	0.50	1.00	57.81	23.75	459.30	3.54	22.40
0.08	610.00	0.50	1.00	57.81	23.13	496.60	3.28	25.20
0.08	700.00	0.50	1.00	64.90	19.38	431.30	3.33	21.30
0.08	790.00	0.50	1.00	40.10	23.13	335.60	3.13	21.10
0.08	880.00	0.50	1.00	47.19	23.75	106.60	2.20	19.50
0.08	970.00	0.50	1.00	36.56	20.00	367.40	1.94	13.60
0.08	1000.00	0.50	1.00	40.10	21.25	344.90	1.54	14.90
0.08	1100.00	0.50	1.00	47.19	22.50	179.80	1.47	9.70
0.08	1200.00	0.50	1.00	50.73	21.52	336.10	1.32	10.30
0.08	1300.00	0.50	1.00	61.36	21.77	296.30	1.22	11.70
0.08	520.00	0.50	1.00	57.81	6.51	330.80	3.54	22.40
0.08	700.00	0.50	1.00	64.90	7.52	427.60	3.93	28.30
0.08	880.00	0.50	1.00	50.73	9.53	360.70	2.20	19.50
0.08	1060.00	0.50	1.00	50.73	25.61	325.20	1.14	7.90
0.08	1200.00	0.50	1.00	54.27	8.54	499.10	1.32	10.30
0.08	160.00	0.50	1.00	50.73	17.50	378.40	4.67	32.80
0.08	250.00	0.50	1.00	36.56	18.75	338.30	3.85	27.60
0.08	340.00	0.50	1.00	40.10	18.75	360.70	3.73	25.90
0.08	430.00	0.50	1.00	43.65	21.88	405.50	4.38	25.80
0.08	520.00	0.50	1.00	50.73	23.75	392.10	3.54	22.40
0.08	610.00	0.50	1.00	57.81	23.13	472.80	3.28	25.20
0.08	700.00	0.50	1.00	61.36	20.00	468.40	3.33	21.30
0.08	790.00	0.50	1.00	36.56	22.50	406.30	3.13	21.10
0.08	880.00	0.50	1.00	47.19	24.38	264.70	2.20	19.50
0.08	970.00	0.50	1.00	33.02	20.00	376.70	1.94	13.60
0.08	1000.00	0.50	1.00	36.56	21.88	361.30	1.54	14.90
0.08	1100.00	0.50	1.00	47.19	23.13	373.30	1.47	9.70
0.08	1200.00	0.50	1.00	50.73	21.67	336.10	1.32	10.30
0.08	1300.00	0.50	1.00	61.36	21.77	296.30	1.22	11.70
0.08	520.00	0.50	1.00	57.81	6.51	330.80	3.54	22.40
0.08	700.00	0.50	1.00	64.90	7.52	427.60	3.93	28.30
0.08	880.00	0.50	1.00	50.73	9.53	360.70	2.20	19.50
0.08	1060.00	0.50	1.00	50.73	25.61	325.20	1.14	7.90
0.08	1200.00	0.50	1.00	54.27	8.54	499.10	1.32	10.30

Table E.4: Complete BPNN training data

E2 BPNN ORIGINAL DATA FOR SURFACE FINISH PREDICTION

Data Specification	Inputs: 7		Outputs: 1		Patterns: 228			
Parameter Name / Data Pattern Number	Feed	Speed	Depth	Tool	Tang.F	Feed.F	Freq	Ry
1	0.03	300.00	0.50	0.00	5.56	4.38	270.00	17.80
2	0.03	300.00	0.50	0.00	3.33	2.50	309.40	17.80
3	0.03	300.00	0.50	0.00	3.33	2.50	302.40	17.80
4	0.03	300.00	0.50	1.00	4.44	5.63	274.10	16.30
5	0.03	300.00	0.50	1.00	3.33	5.00	316.80	16.30
6	0.03	300.00	0.50	1.00	4.44	5.63	324.80	16.30
7	0.04	300.00	0.50	0.00	5.56	7.50	310.60	19.80
8	0.04	300.00	0.50	0.00	3.33	7.50	350.10	19.80
9	0.04	300.00	0.50	0.00	6.67	3.75	254.10	19.80
10	0.04	300.00	0.50	1.00	8.89	5.63	220.50	17.40
11	0.04	300.00	0.50	1.00	8.89	5.63	310.70	17.40
12	0.04	300.00	0.50	1.00	8.89	5.63	260.10	17.40
13	0.05	300.00	0.50	0.00	8.22	6.88	310.60	17.30
14	0.05	300.00	0.50	0.00	8.22	6.25	275.80	17.30
15	0.05	300.00	0.50	0.00	8.22	6.88	320.00	17.30
16	0.05	300.00	0.50	1.00	8.89	6.25	304.00	18.30
17	0.05	300.00	0.50	1.00	10.00	7.50	249.90	18.30
18	0.05	300.00	0.50	1.00	10.00	7.50	280.50	18.30
19	0.06	300.00	0.50	0.00	22.39	8.75	331.40	17.00
20	0.06	300.00	0.50	0.00	18.85	9.38	250.90	17.00
21	0.06	300.00	0.50	0.00	11.76	7.50	335.20	17.00
22	0.06	300.00	0.50	1.00	18.85	7.50	295.40	20.20
23	0.06	300.00	0.50	1.00	18.85	7.50	330.60	20.20
24	0.06	300.00	0.50	1.00	18.85	7.50	320.50	20.20
25	0.08	300.00	0.50	0.00	36.56	8.75	288.60	20.60
26	0.08	300.00	0.50	0.00	29.48	8.75	315.20	20.60
27	0.08	300.00	0.50	0.00	18.85	9.38	320.80	20.60
28	0.08	300.00	0.10	0.00	2.22	2.50	166.30	20.40
29	0.08	300.00	0.20	0.00	5.56	3.75	275.20	21.20
30	0.08	300.00	0.30	0.00	11.76	6.25	244.00	19.40
31	0.08	300.00	0.40	0.00	15.31	6.88	248.20	19.60
32	0.08	300.00	0.50	0.00	43.65	13.75	270.90	22.20
33	0.08	300.00	0.60	0.00	50.73	16.88	272.50	25.60
34	0.08	300.00	0.70	0.00	61.36	19.38	348.20	26.30
35	0.08	300.00	0.80	0.00	75.53	26.25	379.00	28.60
36	0.08	300.00	0.90	0.00	82.61	31.88	365.00	25.00
37	0.08	300.00	1.00	0.00	121.58	39.38	375.00	26.10
38	0.08	300.00	1.10	0.00	139.29	47.50	409.80	27.00
39	0.08	300.00	0.10	0.00	3.33	1.25	164.80	20.40
40	0.08	300.00	0.20	0.00	2.22	3.13	250.60	21.20
41	0.08	300.00	0.30	0.00	7.78	4.38	268.80	19.40
42	0.08	300.00	0.40	0.00	15.31	6.25	244.80	19.60
43	0.08	300.00	0.50	0.00	36.56	13.13	252.10	22.20
44	0.08	300.00	0.60	0.00	43.65	16.25	214.00	25.60
45	0.08	300.00	0.70	0.00	64.90	19.38	241.20	26.30
46	0.08	300.00	0.80	0.00	75.53	23.75	354.20	28.60

Table E.5: BPNN prediction data

47	0.08	300.00	0.90	0.00	86.15	33.13	336.70	25.00
48	0.08	300.00	1.00	0.00	118.03	41.25	404.50	26.10
49	0.08	300.00	1.10	0.00	142.83	50.63	365.00	27.00
50	0.08	300.00	0.10	0.00	0.00	0.00	173.20	20.40
51	0.08	300.00	0.20	0.00	2.22	3.75	245.40	21.20
52	0.08	300.00	0.30	0.00	5.56	6.25	224.40	19.40
53	0.08	300.00	0.40	0.00	8.22	6.25	270.20	19.60
54	0.08	300.00	0.50	0.00	29.48	14.38	253.60	22.20
55	0.08	300.00	0.60	0.00	40.10	16.25	352.00	25.60
56	0.08	300.00	0.70	0.00	57.81	19.38	332.10	26.30
57	0.08	300.00	0.80	0.00	71.98	25.00	370.70	28.60
58	0.08	300.00	0.90	0.00	82.61	33.13	315.30	25.00
59	0.08	300.00	1.00	0.00	118.03	42.50	374.50	26.10
60	0.08	300.00	1.10	0.00	135.75	51.25	360.10	27.00
61	0.08	70.00	0.50	0.00	47.19	23.75	350.40	38.70
62	0.08	160.00	0.50	0.00	36.56	17.50	342.00	28.80
63	0.08	250.00	0.50	0.00	25.93	15.00	344.20	22.60
64	0.08	340.00	0.50	0.00	18.85	13.13	356.90	33.10
65	0.08	430.00	0.50	0.00	15.31	14.38	352.70	15.00
66	0.08	520.00	0.50	0.00	15.31	14.38	204.10	20.80
67	0.08	610.00	0.50	0.00	15.31	12.50	231.40	15.60
68	0.08	700.00	0.50	0.00	11.76	13.13	238.20	17.30
69	0.08	790.00	0.50	0.00	8.22	9.38	299.90	15.80
70	0.08	880.00	0.50	0.00	11.76	9.38	256.40	16.80
71	0.08	970.00	0.50	0.00	22.39	18.75	265.50	14.00
72	0.08	1000.00	0.50	0.00	29.48	20.28	302.90	13.80
73	0.08	1100.00	0.50	0.00	36.56	21.39	272.30	16.50
74	0.08	1200.00	0.50	0.00	40.10	17.88	281.70	11.20
75	0.08	1300.00	0.50	0.00	29.48	20.86	245.50	9.90
76	0.08	70.00	0.50	0.00	47.19	23.75	374.90	38.70
77	0.08	160.00	0.50	0.00	36.56	16.88	360.00	28.80
78	0.08	250.00	0.50	0.00	25.93	15.63	263.30	22.60
79	0.08	340.00	0.50	0.00	22.39	13.75	364.20	33.10
80	0.08	430.00	0.50	0.00	15.31	15.00	317.20	15.00
81	0.08	520.00	0.50	0.00	15.31	15.63	243.90	20.80
82	0.08	610.00	0.50	0.00	11.76	12.50	226.10	15.60
83	0.08	700.00	0.50	0.00	11.76	11.88	279.70	17.30
84	0.08	790.00	0.50	0.00	8.22	8.75	200.20	15.80
85	0.08	880.00	0.50	0.00	11.76	9.38	305.00	16.80
86	0.08	970.00	0.50	0.00	29.48	18.75	266.90	14.00
87	0.08	1000.00	0.50	0.00	25.93	20.74	261.90	13.80
88	0.08	1100.00	0.50	0.00	33.02	21.94	300.30	16.50
89	0.08	1200.00	0.50	0.00	43.65	18.02	281.40	11.20
90	0.08	1300.00	0.50	0.00	29.48	20.89	210.80	9.90
91	0.08	70.00	0.50	0.00	50.73	23.75	330.60	38.70
92	0.08	160.00	0.50	0.00	36.56	18.13	363.20	28.80
93	0.08	250.00	0.50	0.00	33.02	15.63	249.80	22.60
94	0.08	340.00	0.50	0.00	22.39	13.13	240.20	33.10
95	0.08	430.00	0.50	0.00	15.31	15.00	320.80	15.00
96	0.08	520.00	0.50	0.00	15.31	15.00	260.70	20.80

Table E.6: BPNN prediction data

97	0.08	610.00	0.50	0.00	11.76	12.50	253.60	15.60
98	0.08	700.00	0.50	0.00	11.76	13.75	228.40	17.30
99	0.08	790.00	0.50	0.00	11.76	10.00	200.10	15.80
100	0.08	880.00	0.50	0.00	11.76	10.63	240.00	16.80
101	0.08	970.00	0.50	0.00	29.48	18.13	229.80	14.00
102	0.08	1000.00	0.50	0.00	22.39	20.28	363.40	13.80
103	0.08	1100.00	0.50	0.00	33.02	21.39	310.30	16.50
104	0.08	1200.00	0.50	0.00	40.10	17.88	345.00	11.20
105	0.08	1300.00	0.50	0.00	29.48	20.86	290.40	9.90
106	0.08	300.00	0.50	1.00	36.56	15.63	357.30	19.90
107	0.08	300.00	0.50	1.00	40.10	15.00	384.60	19.90
108	0.08	300.00	0.50	1.00	29.48	15.00	310.20	19.90
109	0.08	300.00	0.10	1.00	5.56	3.13	359.70	19.60
110	0.08	300.00	0.20	1.00	7.78	3.75	345.60	27.30
111	0.08	300.00	0.30	1.00	15.31	10.00	259.90	29.00
112	0.08	300.00	0.40	1.00	22.39	13.13	304.00	24.20
113	0.08	300.00	0.50	1.00	22.39	15.00	368.10	23.00
114	0.08	300.00	0.60	1.00	36.56	25.63	368.00	20.90
115	0.08	300.00	0.70	1.00	40.10	28.75	374.70	20.00
116	0.08	300.00	0.80	1.00	57.81	30.63	378.60	22.90
117	0.08	300.00	0.90	1.00	68.44	36.25	380.60	21.90
118	0.08	300.00	1.00	1.00	82.61	39.38	381.40	29.70
119	0.08	300.00	1.10	1.00	89.70	41.88	376.80	34.40
120	0.08	300.00	0.10	1.00	6.67	3.13	262.00	19.60
121	0.08	300.00	0.20	1.00	7.78	3.75	366.50	27.30
122	0.08	300.00	0.30	1.00	15.31	9.38	346.90	29.00
123	0.08	300.00	0.40	1.00	18.85	11.25	297.80	24.20
124	0.08	300.00	0.50	1.00	22.39	13.13	297.20	23.00
125	0.08	300.00	0.60	1.00	29.48	25.63	376.10	20.90
126	0.08	300.00	0.70	1.00	29.48	30.00	377.60	20.00
127	0.08	300.00	0.80	1.00	64.90	31.25	379.40	22.90
128	0.08	300.00	0.90	1.00	64.90	37.50	374.60	21.90
129	0.08	300.00	1.00	1.00	82.61	39.38	381.50	29.70
130	0.08	300.00	1.10	1.00	89.70	43.13	370.60	34.40
131	0.08	300.00	0.10	1.00	5.56	3.13	282.30	19.60
132	0.08	300.00	0.20	1.00	7.78	3.13	288.80	27.30
133	0.08	300.00	0.30	1.00	11.76	10.63	316.50	29.00
134	0.08	300.00	0.40	1.00	15.31	10.00	293.90	24.20
135	0.08	300.00	0.50	1.00	25.93	13.75	252.20	23.00
136	0.08	300.00	0.60	1.00	22.39	26.25	377.70	20.90
137	0.08	300.00	0.70	1.00	36.56	29.38	357.90	20.00
138	0.08	300.00	0.80	1.00	57.81	31.25	379.70	22.90
139	0.08	300.00	0.90	1.00	61.36	38.13	350.30	21.90
140	0.08	300.00	1.00	1.00	79.07	39.38	379.40	29.70
141	0.08	300.00	1.10	1.00	89.70	43.13	368.60	34.40
142	0.08	160.00	0.50	1.00	64.90	19.38	370.60	32.80
143	0.08	250.00	0.50	1.00	54.27	18.75	359.50	27.60
144	0.08	340.00	0.50	1.00	54.27	19.38	398.20	25.90
145	0.08	430.00	0.50	1.00	54.27	21.25	366.20	25.80
146	0.08	520.00	0.50	1.00	61.36	23.75	452.90	22.40

Table E.7: BPNN prediction data

147	0.08	610.00	0.50	1.00	61.36	23.13	485.70	25.20
148	0.08	700.00	0.50	1.00	64.90	19.38	154.40	21.30
149	0.08	790.00	0.50	1.00	43.65	22.50	390.10	21.10
150	0.08	880.00	0.50	1.00	50.73	24.38	499.00	19.50
151	0.08	970.00	0.50	1.00	40.10	20.00	376.20	13.60
152	0.08	1000.00	0.50	1.00	43.65	21.88	362.30	14.90
153	0.08	1100.00	0.50	1.00	54.27	22.50	373.30	9.70
154	0.08	1200.00	0.50	1.00	54.27	21.69	317.50	10.30
155	0.08	1300.00	0.50	1.00	61.36	21.77	296.30	11.70
156	0.08	520.00	0.50	1.00	57.81	6.51	330.80	22.40
157	0.08	700.00	0.50	1.00	64.90	7.52	427.60	28.30
158	0.08	880.00	0.50	1.00	50.73	9.53	360.70	19.50
159	0.08	1060.00	0.50	1.00	50.73	25.61	325.20	7.90
160	0.08	1200.00	0.50	1.00	54.27	8.54	499.10	10.30
161	0.08	160.00	0.50	1.00	57.81	18.13	365.70	32.80
162	0.08	250.00	0.50	1.00	50.73	18.75	349.60	27.60
163	0.08	340.00	0.50	1.00	47.19	19.38	408.70	25.90
164	0.08	430.00	0.50	1.00	47.19	21.88	265.50	25.80
165	0.08	520.00	0.50	1.00	57.81	23.75	459.30	22.40
166	0.08	610.00	0.50	1.00	57.81	23.13	496.60	25.20
167	0.08	700.00	0.50	1.00	64.90	19.38	431.30	21.30
168	0.08	790.00	0.50	1.00	40.10	23.13	335.60	21.10
169	0.08	880.00	0.50	1.00	47.19	23.75	106.60	19.50
170	0.08	970.00	0.50	1.00	36.56	20.00	367.40	13.60
171	0.08	1000.00	0.50	1.00	40.10	21.25	344.90	14.90
172	0.08	1100.00	0.50	1.00	47.19	22.50	179.80	9.70
173	0.08	1200.00	0.50	1.00	50.73	21.52	336.10	10.30
174	0.08	1300.00	0.50	1.00	61.36	21.77	296.30	11.70
175	0.08	520.00	0.50	1.00	57.81	6.51	330.80	22.40
176	0.08	700.00	0.50	1.00	64.90	7.52	427.60	28.30
177	0.08	880.00	0.50	1.00	50.73	9.53	360.70	19.50
178	0.08	1060.00	0.50	1.00	50.73	25.61	325.20	7.90
179	0.08	1200.00	0.50	1.00	54.27	8.54	499.10	10.30
180	0.08	160.00	0.50	1.00	50.73	17.50	378.40	32.80
181	0.08	250.00	0.50	1.00	36.56	18.75	338.30	27.60
182	0.08	340.00	0.50	1.00	40.10	18.75	360.70	25.90
183	0.08	430.00	0.50	1.00	43.65	21.88	405.50	25.80
184	0.08	520.00	0.50	1.00	50.73	23.75	392.10	22.40
185	0.08	610.00	0.50	1.00	57.81	23.13	472.80	25.20
186	0.08	700.00	0.50	1.00	61.36	20.00	468.40	21.30
187	0.08	790.00	0.50	1.00	36.56	22.50	406.30	21.10
188	0.08	880.00	0.50	1.00	47.19	24.38	264.70	19.50
189	0.08	970.00	0.50	1.00	33.02	20.00	376.70	13.60
190	0.08	1000.00	0.50	1.00	36.56	21.88	361.30	14.90
191	0.08	1100.00	0.50	1.00	47.19	23.13	373.30	9.70
192	0.08	1200.00	0.50	1.00	50.73	21.67	336.10	10.30
193	0.08	1300.00	0.50	1.00	61.36	21.77	296.30	11.70
194	0.08	520.00	0.50	1.00	57.81	6.51	330.80	22.40
195	0.08	700.00	0.50	1.00	64.90	7.52	427.60	28.30
196	0.08	880.00	0.50	1.00	50.73	9.53	360.70	19.50

Table E.8: BPNN prediction data

197	0.08	1060.00	0.50	1.00	50.73	25.61	325.20	7.90
198	0.08	1200.00	0.50	1.00	54.27	8.54	499.10	10.30
199	0.10	300.00	0.50	0.00	33.02	12.50	292.00	18.90
200	0.10	300.00	0.50	0.00	29.48	11.25	369.40	18.90
201	0.10	300.00	0.50	0.00	25.93	10.63	320.10	18.90
202	0.10	300.00	0.50	1.00	40.10	18.75	342.00	24.80
203	0.10	300.00	0.50	1.00	40.10	18.75	367.10	24.80
204	0.10	300.00	0.50	1.00	40.10	18.75	345.90	24.80
205	0.12	300.00	0.50	0.00	43.65	13.75	335.60	23.20
206	0.12	300.00	0.50	0.00	43.65	13.13	335.90	23.20
207	0.12	300.00	0.50	0.00	40.10	13.75	279.80	23.20
208	0.12	300.00	0.50	1.00	33.02	20.63	360.60	27.00
209	0.12	300.00	0.50	1.00	29.48	20.00	369.80	27.00
210	0.12	300.00	0.50	1.00	22.39	19.38	385.70	27.00
211	0.16	300.00	0.50	0.00	64.90	16.25	328.30	19.40
212	0.16	300.00	0.50	0.00	64.90	15.63	321.10	19.40
213	0.16	300.00	0.50	0.00	64.90	16.25	320.50	19.40
214	0.16	300.00	0.50	1.00	61.36	21.88	267.80	23.50
215	0.16	300.00	0.50	1.00	61.36	22.50	326.30	23.50
216	0.16	300.00	0.50	1.00	57.81	23.75	357.60	23.50
217	0.20	300.00	0.50	0.00	135.75	21.25	361.70	22.60
218	0.20	300.00	0.50	0.00	132.20	21.88	361.60	22.60
219	0.20	300.00	0.50	0.00	132.20	21.88	356.20	22.60
220	0.20	300.00	0.50	1.00	96.78	26.88	378.00	28.40
221	0.20	300.00	0.50	1.00	100.32	26.88	328.40	28.40
222	0.20	300.00	0.50	1.00	96.78	26.88	381.00	28.40
223	0.25	300.00	0.50	0.00	231.39	26.38	357.20	21.90
224	0.25	300.00	0.50	0.00	224.30	27.63	377.90	21.90
225	0.25	300.00	0.50	0.00	224.30	27.00	360.70	21.90
226	0.25	300.00	0.50	1.00	142.83	31.25	336.30	27.70
227	0.25	300.00	0.50	1.00	142.83	32.50	339.30	27.70
228	0.25	300.00	0.50	1.00	142.83	31.88	312.00	27.70
min.	0.03	70.00	0.10	0.00	0.00	0.00	106.60	7.90
max.	0.25	1300.00	1.10	1.00	231.39	51.25	499.10	38.70

Table E.9: BPNN prediction data

E3 BPNN NORMALISED DATA FOR SURFACE FINISH PREDICTION

Parameter Name / Data Pattern Number	Feed	Speed	Depth	Tool	Tang.F	Feed.F	Freq	Ry
1	0.00000	0.18699	0.40000	0.00000	0.02401	0.08537	0.41631	0.32143
2	0.00000	0.18699	0.40000	0.00000	0.01441	0.04878	0.51669	0.32143
3	0.00000	0.18699	0.40000	0.00000	0.01441	0.04878	0.49885	0.32143
4	0.00000	0.18699	0.40000	1.00000	0.01921	0.10976	0.42675	0.27273
5	0.00000	0.18699	0.40000	1.00000	0.01441	0.09756	0.53554	0.27273
6	0.00000	0.18699	0.40000	1.00000	0.01921	0.10976	0.55592	0.27273
7	0.04545	0.18699	0.40000	0.00000	0.02401	0.14634	0.51975	0.38636
8	0.04545	0.18699	0.40000	0.00000	0.01441	0.14634	0.62038	0.38636
9	0.04545	0.18699	0.40000	0.00000	0.02881	0.07317	0.37580	0.38636
10	0.04545	0.18699	0.40000	1.00000	0.03842	0.10976	0.29019	0.30844
11	0.04545	0.18699	0.40000	1.00000	0.03842	0.10976	0.52000	0.30844
12	0.04545	0.18699	0.40000	1.00000	0.03842	0.10976	0.39108	0.30844
13	0.09091	0.18699	0.40000	0.00000	0.03553	0.13415	0.51975	0.30519
14	0.09091	0.18699	0.40000	0.00000	0.03553	0.12195	0.43108	0.30519
15	0.09091	0.18699	0.40000	0.00000	0.03553	0.13415	0.54369	0.30519
16	0.09091	0.18699	0.40000	1.00000	0.03842	0.12195	0.50293	0.33766
17	0.09091	0.18699	0.40000	1.00000	0.04322	0.14634	0.36510	0.33766
18	0.09091	0.18699	0.40000	1.00000	0.04322	0.14634	0.44306	0.33766
19	0.13636	0.18699	0.40000	0.00000	0.09677	0.17073	0.57274	0.29545
20	0.13636	0.18699	0.40000	0.00000	0.08146	0.18293	0.36764	0.29545
21	0.13636	0.18699	0.40000	0.00000	0.05084	0.14634	0.58242	0.29545
22	0.13636	0.18699	0.40000	1.00000	0.08146	0.14634	0.48102	0.39935
23	0.13636	0.18699	0.40000	1.00000	0.08146	0.14634	0.57070	0.39935
24	0.13636	0.18699	0.40000	1.00000	0.08146	0.14634	0.54497	0.39935
25	0.22727	0.18699	0.40000	0.00000	0.15801	0.17073	0.46369	0.41234
26	0.22727	0.18699	0.40000	0.00000	0.12739	0.17073	0.53146	0.41234
27	0.22727	0.18699	0.40000	0.00000	0.08146	0.18293	0.54573	0.41234
28	0.22727	0.18699	0.00000	0.00000	0.00960	0.04878	0.15210	0.40584
29	0.22727	0.18699	0.10000	0.00000	0.02401	0.07317	0.42955	0.43182
30	0.22727	0.18699	0.20000	0.00000	0.05084	0.12195	0.35006	0.37338
31	0.22727	0.18699	0.30000	0.00000	0.06615	0.13415	0.36076	0.37987
32	0.22727	0.18699	0.40000	0.00000	0.18862	0.26829	0.41860	0.46429
33	0.22727	0.18699	0.50000	0.00000	0.21924	0.32927	0.42268	0.57468
34	0.22727	0.18699	0.60000	0.00000	0.26517	0.37805	0.61554	0.59740
35	0.22727	0.18699	0.70000	0.00000	0.32640	0.51220	0.69401	0.67208
36	0.22727	0.18699	0.80000	0.00000	0.35702	0.62195	0.65834	0.55519
37	0.22727	0.18699	0.90000	0.00000	0.52542	0.76829	0.68382	0.59091
38	0.22727	0.18699	1.00000	0.00000	0.60197	0.92683	0.77248	0.62013
39	0.22727	0.18699	0.00000	0.00000	0.01441	0.02439	0.14828	0.40584
40	0.22727	0.18699	0.10000	0.00000	0.00960	0.06098	0.36688	0.43182
41	0.22727	0.18699	0.20000	0.00000	0.03361	0.08537	0.41325	0.37338
42	0.22727	0.18699	0.30000	0.00000	0.06615	0.12195	0.35210	0.37987
43	0.22727	0.18699	0.40000	0.00000	0.15801	0.25610	0.37070	0.46429
44	0.22727	0.18699	0.50000	0.00000	0.18862	0.31707	0.27363	0.57468
45	0.22727	0.18699	0.60000	0.00000	0.28048	0.37805	0.34293	0.59740
46	0.22727	0.18699	0.70000	0.00000	0.32640	0.46341	0.63083	0.67208
47	0.22727	0.18699	0.80000	0.00000	0.37233	0.64634	0.58624	0.55519
48	0.22727	0.18699	0.90000	0.00000	0.51011	0.80488	0.75898	0.59091
49	0.22727	0.18699	1.00000	0.00000	0.61728	0.98780	0.65834	0.62013

Table E.10: BPNN normalised prediction data

50	0.22727	0.18699	0.00000	0.00000	0.00000	0.00000	0.16968	0.40584
51	0.22727	0.18699	0.10000	0.00000	0.00960	0.07317	0.35363	0.43182
52	0.22727	0.18699	0.20000	0.00000	0.02401	0.12195	0.30013	0.37338
53	0.22727	0.18699	0.30000	0.00000	0.03553	0.12195	0.41682	0.37987
54	0.22727	0.18699	0.40000	0.00000	0.12739	0.28049	0.37452	0.46429
55	0.22727	0.18699	0.50000	0.00000	0.17331	0.31707	0.62522	0.57468
56	0.22727	0.18699	0.60000	0.00000	0.24986	0.37805	0.57452	0.59740
57	0.22727	0.18699	0.70000	0.00000	0.31110	0.48780	0.67287	0.67208
58	0.22727	0.18699	0.80000	0.00000	0.35702	0.64634	0.53172	0.55519
59	0.22727	0.18699	0.90000	0.00000	0.51011	0.82927	0.68255	0.59091
60	0.22727	0.18699	1.00000	0.00000	0.58666	1.00000	0.64586	0.62013
61	0.22727	0.00000	0.40000	0.00000	0.20393	0.46341	0.62115	1.00000
62	0.22727	0.07317	0.40000	0.00000	0.15801	0.34146	0.59975	0.67857
63	0.22727	0.14634	0.40000	0.00000	0.11208	0.29268	0.60535	0.47727
64	0.22727	0.21951	0.40000	0.00000	0.08146	0.25610	0.63771	0.81818
65	0.22727	0.29268	0.40000	0.00000	0.06615	0.28049	0.62701	0.23052
66	0.22727	0.36585	0.40000	0.00000	0.06615	0.28049	0.24841	0.41883
67	0.22727	0.43902	0.40000	0.00000	0.06615	0.24390	0.31796	0.25000
68	0.22727	0.51220	0.40000	0.00000	0.05084	0.25610	0.33529	0.30519
69	0.22727	0.58537	0.40000	0.00000	0.03553	0.18293	0.49248	0.25649
70	0.22727	0.65854	0.40000	0.00000	0.05084	0.18293	0.38166	0.28896
71	0.22727	0.73171	0.40000	0.00000	0.09677	0.36585	0.40484	0.19805
72	0.22727	0.75610	0.40000	0.00000	0.12739	0.39573	0.50013	0.19156
73	0.22727	0.83740	0.40000	0.00000	0.15801	0.41732	0.42217	0.27922
74	0.22727	0.91870	0.40000	0.00000	0.17331	0.34890	0.44611	0.10714
75	0.22727	1.00000	0.40000	0.00000	0.12739	0.40695	0.35389	0.06494
76	0.22727	0.00000	0.40000	0.00000	0.20393	0.46341	0.68357	1.00000
77	0.22727	0.07317	0.40000	0.00000	0.15801	0.32927	0.64561	0.67857
78	0.22727	0.14634	0.40000	0.00000	0.11208	0.30488	0.39924	0.47727
79	0.22727	0.21951	0.40000	0.00000	0.09677	0.26829	0.65631	0.81818
80	0.22727	0.29268	0.40000	0.00000	0.06615	0.29268	0.53656	0.23052
81	0.22727	0.36585	0.40000	0.00000	0.06615	0.30488	0.34981	0.41883
82	0.22727	0.43902	0.40000	0.00000	0.05084	0.24390	0.30446	0.25000
83	0.22727	0.51220	0.40000	0.00000	0.05084	0.23171	0.44102	0.30519
84	0.22727	0.58537	0.40000	0.00000	0.03553	0.17073	0.23847	0.25649
85	0.22727	0.65854	0.40000	0.00000	0.05084	0.18293	0.50548	0.28896
86	0.22727	0.73171	0.40000	0.00000	0.12739	0.36585	0.40841	0.19805
87	0.22727	0.75610	0.40000	0.00000	0.11208	0.40476	0.39567	0.19156
88	0.22727	0.83740	0.40000	0.00000	0.14270	0.42805	0.49350	0.27922
89	0.22727	0.91870	0.40000	0.00000	0.18862	0.35159	0.44535	0.10714
90	0.22727	1.00000	0.40000	0.00000	0.12739	0.40768	0.26548	0.06494
91	0.22727	0.00000	0.40000	0.00000	0.21924	0.46341	0.57070	1.00000
92	0.22727	0.07317	0.40000	0.00000	0.15801	0.35366	0.65376	0.67857
93	0.22727	0.14634	0.40000	0.00000	0.14270	0.30488	0.36484	0.47727
94	0.22727	0.21951	0.40000	0.00000	0.09677	0.25610	0.34038	0.81818
95	0.22727	0.29268	0.40000	0.00000	0.06615	0.29268	0.54573	0.23052
96	0.22727	0.36585	0.40000	0.00000	0.06615	0.29268	0.39261	0.41883
97	0.22727	0.43902	0.40000	0.00000	0.05084	0.24390	0.37452	0.25000
98	0.22727	0.51220	0.40000	0.00000	0.05084	0.26829	0.31032	0.30519
99	0.22727	0.58537	0.40000	0.00000	0.05084	0.19512	0.23822	0.25649

Table E.11: BPNN normalised prediction data

100	0.22727	0.65854	0.40000	0.00000	0.05084	0.20732	0.33987	0.28896
101	0.22727	0.73171	0.40000	0.00000	0.12739	0.35366	0.31389	0.19805
102	0.22727	0.75610	0.40000	0.00000	0.09677	0.39573	0.65427	0.19156
103	0.22727	0.83740	0.40000	0.00000	0.14270	0.41732	0.51898	0.27922
104	0.22727	0.91870	0.40000	0.00000	0.17331	0.34890	0.60739	0.10714
105	0.22727	1.00000	0.40000	0.00000	0.12739	0.40695	0.46828	0.06494
106	0.22727	0.18699	0.40000	1.00000	0.15801	0.30488	0.63873	0.38961
107	0.22727	0.18699	0.40000	1.00000	0.17331	0.29268	0.70828	0.38961
108	0.22727	0.18699	0.40000	1.00000	0.12739	0.29268	0.51873	0.38961
109	0.22727	0.18699	0.00000	1.00000	0.02401	0.06098	0.64484	0.37987
110	0.22727	0.18699	0.10000	1.00000	0.03361	0.07317	0.60892	0.62987
111	0.22727	0.18699	0.20000	1.00000	0.06615	0.19512	0.39057	0.68506
112	0.22727	0.18699	0.30000	1.00000	0.09677	0.25610	0.50293	0.52922
113	0.22727	0.18699	0.40000	1.00000	0.09677	0.29268	0.66624	0.49026
114	0.22727	0.18699	0.50000	1.00000	0.15801	0.50000	0.66599	0.42208
115	0.22727	0.18699	0.60000	1.00000	0.17331	0.56098	0.68306	0.39286
116	0.22727	0.18699	0.70000	1.00000	0.24986	0.59756	0.69299	0.48701
117	0.22727	0.18699	0.80000	1.00000	0.29579	0.70732	0.69809	0.45455
118	0.22727	0.18699	0.90000	1.00000	0.35702	0.76829	0.70013	0.70779
119	0.22727	0.18699	1.00000	1.00000	0.38764	0.81707	0.68841	0.86039
120	0.22727	0.18699	0.00000	1.00000	0.02881	0.06098	0.39592	0.37987
121	0.22727	0.18699	0.10000	1.00000	0.03361	0.07317	0.66217	0.62987
122	0.22727	0.18699	0.20000	1.00000	0.06615	0.18293	0.61223	0.68506
123	0.22727	0.18699	0.30000	1.00000	0.08146	0.21951	0.48713	0.52922
124	0.22727	0.18699	0.40000	1.00000	0.09677	0.25610	0.48561	0.49026
125	0.22727	0.18699	0.50000	1.00000	0.12739	0.50000	0.68662	0.42208
126	0.22727	0.18699	0.60000	1.00000	0.12739	0.58537	0.69045	0.39286
127	0.22727	0.18699	0.70000	1.00000	0.28048	0.60976	0.69503	0.48701
128	0.22727	0.18699	0.80000	1.00000	0.28048	0.73171	0.68280	0.45455
129	0.22727	0.18699	0.90000	1.00000	0.35702	0.76829	0.70038	0.70779
130	0.22727	0.18699	1.00000	1.00000	0.38764	0.84146	0.67261	0.86039
131	0.22727	0.18699	0.00000	1.00000	0.02401	0.06098	0.44764	0.37987
132	0.22727	0.18699	0.10000	1.00000	0.03361	0.06098	0.46420	0.62987
133	0.22727	0.18699	0.20000	1.00000	0.05084	0.20732	0.53478	0.68506
134	0.22727	0.18699	0.30000	1.00000	0.06615	0.19512	0.47720	0.52922
135	0.22727	0.18699	0.40000	1.00000	0.11208	0.26829	0.37096	0.49026
136	0.22727	0.18699	0.50000	1.00000	0.09677	0.51220	0.69070	0.42208
137	0.22727	0.18699	0.60000	1.00000	0.15801	0.57317	0.64025	0.39286
138	0.22727	0.18699	0.70000	1.00000	0.24986	0.60976	0.69580	0.48701
139	0.22727	0.18699	0.80000	1.00000	0.26517	0.74390	0.62089	0.45455
140	0.22727	0.18699	0.90000	1.00000	0.34171	0.76829	0.69503	0.70779
141	0.22727	0.18699	1.00000	1.00000	0.38764	0.84146	0.66752	0.86039
142	0.22727	0.07317	0.40000	1.00000	0.28048	0.37805	0.67261	0.80844
143	0.22727	0.14634	0.40000	1.00000	0.23455	0.36585	0.64433	0.63961
144	0.22727	0.21951	0.40000	1.00000	0.23455	0.37805	0.74293	0.58442
145	0.22727	0.29268	0.40000	1.00000	0.23455	0.41463	0.66140	0.58117
146	0.22727	0.36585	0.40000	1.00000	0.26517	0.46341	0.88229	0.47078
147	0.22727	0.43902	0.40000	1.00000	0.26517	0.45122	0.96586	0.56169
148	0.22727	0.51220	0.40000	1.00000	0.28048	0.37805	0.12178	0.43506
149	0.22727	0.58537	0.40000	1.00000	0.18862	0.43902	0.72229	0.42857

Table E.12: BPNN normalised prediction data

150	0.22727	0.65854	0.40000	1.00000	0.21924	0.47561	0.99975	0.37662
151	0.22727	0.73171	0.40000	1.00000	0.17331	0.39024	0.68688	0.18506
152	0.22727	0.75610	0.40000	1.00000	0.18862	0.42683	0.65146	0.22727
153	0.22727	0.83740	0.40000	1.00000	0.23455	0.43902	0.67949	0.05844
154	0.22727	0.91870	0.40000	1.00000	0.23455	0.42317	0.53732	0.07792
155	0.22727	1.00000	0.40000	1.00000	0.26517	0.42476	0.48331	0.12338
156	0.22727	0.36585	0.40000	1.00000	0.24986	0.12707	0.57121	0.47078
157	0.22727	0.51220	0.40000	1.00000	0.28048	0.14671	0.81783	0.66234
158	0.22727	0.65854	0.40000	1.00000	0.21924	0.18598	0.64739	0.37662
159	0.22727	0.80488	0.40000	1.00000	0.21924	0.49976	0.55694	0.00000
160	0.22727	0.91870	0.40000	1.00000	0.23455	0.16671	1.00000	0.07792
161	0.22727	0.07317	0.40000	1.00000	0.24986	0.35366	0.66013	0.80844
162	0.22727	0.14634	0.40000	1.00000	0.21924	0.36585	0.61911	0.63961
163	0.22727	0.21951	0.40000	1.00000	0.20393	0.37805	0.76968	0.58442
164	0.22727	0.29268	0.40000	1.00000	0.20393	0.42683	0.40484	0.58117
165	0.22727	0.36585	0.40000	1.00000	0.24986	0.46341	0.89860	0.47078
166	0.22727	0.43902	0.40000	1.00000	0.24986	0.45122	0.99363	0.56169
167	0.22727	0.51220	0.40000	1.00000	0.28048	0.37805	0.82726	0.43506
168	0.22727	0.58537	0.40000	1.00000	0.17331	0.45122	0.58344	0.42857
169	0.22727	0.65854	0.40000	1.00000	0.20393	0.46341	0.00000	0.37662
170	0.22727	0.73171	0.40000	1.00000	0.15801	0.39024	0.66446	0.18506
171	0.22727	0.75610	0.40000	1.00000	0.17331	0.41463	0.60713	0.22727
172	0.22727	0.83740	0.40000	1.00000	0.20393	0.43902	0.18650	0.05844
173	0.22727	0.91870	0.40000	1.00000	0.21924	0.41988	0.58471	0.07792
174	0.22727	1.00000	0.40000	1.00000	0.26517	0.42476	0.48331	0.12338
175	0.22727	0.36585	0.40000	1.00000	0.24986	0.12707	0.57121	0.47078
176	0.22727	0.51220	0.40000	1.00000	0.28048	0.14671	0.81783	0.66234
177	0.22727	0.65854	0.40000	1.00000	0.21924	0.18598	0.64739	0.37662
178	0.22727	0.80488	0.40000	1.00000	0.21924	0.49976	0.55694	0.00000
179	0.22727	0.91870	0.40000	1.00000	0.23455	0.16671	1.00000	0.07792
180	0.22727	0.07317	0.40000	1.00000	0.21924	0.34146	0.69248	0.80844
181	0.22727	0.14634	0.40000	1.00000	0.15801	0.36585	0.59032	0.63961
182	0.22727	0.21951	0.40000	1.00000	0.17331	0.36585	0.64739	0.58442
183	0.22727	0.29268	0.40000	1.00000	0.18862	0.42683	0.76153	0.58117
184	0.22727	0.36585	0.40000	1.00000	0.21924	0.46341	0.72739	0.47078
185	0.22727	0.43902	0.40000	1.00000	0.24986	0.45122	0.93299	0.56169
186	0.22727	0.51220	0.40000	1.00000	0.26517	0.39024	0.92178	0.43506
187	0.22727	0.58537	0.40000	1.00000	0.15801	0.43902	0.76357	0.42857
188	0.22727	0.65854	0.40000	1.00000	0.20393	0.47561	0.40280	0.37662
189	0.22727	0.73171	0.40000	1.00000	0.14270	0.39024	0.68815	0.18506
190	0.22727	0.75610	0.40000	1.00000	0.15801	0.42683	0.64892	0.22727
191	0.22727	0.83740	0.40000	1.00000	0.20393	0.45122	0.67949	0.05844
192	0.22727	0.91870	0.40000	1.00000	0.21924	0.42277	0.58471	0.07792
193	0.22727	1.00000	0.40000	1.00000	0.26517	0.42476	0.48331	0.12338
194	0.22727	0.36585	0.40000	1.00000	0.24986	0.12707	0.57121	0.47078
195	0.22727	0.51220	0.40000	1.00000	0.28048	0.14671	0.81783	0.66234
196	0.22727	0.65854	0.40000	1.00000	0.21924	0.18598	0.64739	0.37662
197	0.22727	0.80488	0.40000	1.00000	0.21924	0.49976	0.55694	0.00000
198	0.22727	0.91870	0.40000	1.00000	0.23455	0.16671	1.00000	0.07792
199	0.31818	0.18699	0.40000	0.00000	0.14270	0.24390	0.47236	0.35714

Table E.13: BPNN normalised prediction data

200	0.31818	0.18699	0.40000	0.00000	0.12739	0.21951	0.66955	0.35714
201	0.31818	0.18699	0.40000	0.00000	0.11208	0.20732	0.54395	0.35714
202	0.31818	0.18699	0.40000	1.00000	0.17331	0.36585	0.59975	0.54870
203	0.31818	0.18699	0.40000	1.00000	0.17331	0.36585	0.66369	0.54870
204	0.31818	0.18699	0.40000	1.00000	0.17331	0.36585	0.60968	0.54870
205	0.40909	0.18699	0.40000	0.00000	0.18862	0.26829	0.58344	0.49675
206	0.40909	0.18699	0.40000	0.00000	0.18862	0.25610	0.58420	0.49675
207	0.40909	0.18699	0.40000	0.00000	0.17331	0.26829	0.44127	0.49675
208	0.40909	0.18699	0.40000	1.00000	0.14270	0.40244	0.64713	0.62013
209	0.40909	0.18699	0.40000	1.00000	0.12739	0.39024	0.67057	0.62013
210	0.40909	0.18699	0.40000	1.00000	0.09677	0.37805	0.71108	0.62013
211	0.59091	0.18699	0.40000	0.00000	0.28048	0.31707	0.56484	0.37338
212	0.59091	0.18699	0.40000	0.00000	0.28048	0.30488	0.54650	0.37338
213	0.59091	0.18699	0.40000	0.00000	0.28048	0.31707	0.54497	0.37338
214	0.59091	0.18699	0.40000	1.00000	0.26517	0.42683	0.41070	0.50649
215	0.59091	0.18699	0.40000	1.00000	0.26517	0.43902	0.55975	0.50649
216	0.59091	0.18699	0.40000	1.00000	0.24986	0.46341	0.63949	0.50649
217	0.77273	0.18699	0.40000	0.00000	0.58666	0.41463	0.64994	0.47727
218	0.77273	0.18699	0.40000	0.00000	0.57135	0.42683	0.64968	0.47727
219	0.77273	0.18699	0.40000	0.00000	0.57135	0.42683	0.63592	0.47727
220	0.77273	0.18699	0.40000	1.00000	0.41826	0.52439	0.69146	0.66558
221	0.77273	0.18699	0.40000	1.00000	0.43357	0.52439	0.56510	0.66558
222	0.77273	0.18699	0.40000	1.00000	0.41826	0.52439	0.69911	0.66558
223	1.00000	0.18699	0.40000	0.00000	1.00000	0.51463	0.63847	0.45455
224	1.00000	0.18699	0.40000	0.00000	0.96938	0.53902	0.69121	0.45455
225	1.00000	0.18699	0.40000	0.00000	0.96938	0.52683	0.64739	0.45455
226	1.00000	0.18699	0.40000	1.00000	0.61728	0.60976	0.58522	0.64286
227	1.00000	0.18699	0.40000	1.00000	0.61728	0.63415	0.59287	0.64286
228	1.00000	0.18699	0.40000	1.00000	0.61728	0.62195	0.52331	0.64286
min.	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
max.	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000

Table E.14: BPNN normalised prediction data